ระบบสืบค้นข้อมูลภาษาไทยในอินเทอร์เน็ต

Thai Search Engine

โดย

นายอรรถพล อาจไชยธร

รหัส 44067093

อาจารย์ที่ปรึกษา

ผศ.ดร.โชติพัชร์ ภรณวลัย

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2545

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	ระบบสืบค้นข้อมูลภาษาไทยในอินเทอร์เน็ต
นักศึกษา	นายอรรถพล อาจไชยธร
อาจารย์ที่ปรึกษา	ผศ.ดร.โชติพัชร์ ภรณวลัย
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโยลีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2545

บทคัดย่อ

ปริมาณของข้อมูลบนเครือข่ายเวิลด์ไวด์เว็บมีจำนวนเพิ่มมากขึ้นอย่างรวดเร็ว เช่นเดียวกันกับจำนวนของผู้ใช้งานอินเทอร์เน็ตที่ขาดประสบการณ์ในการค้นหาข้อมูล ผู้ใช้งานอินเทอร์เน็ตชาวไทยส่วนมากมักนิยมค้นหาข้อมูลบนเครือข่ายเวิลด์ไวด์เว็บโดยใช้ web directory หรือ search engine ซึ่งในการค้นหาข้อมูลโดยใช้ search engine นั้นจะมีปัญหาหนึ่งที่เกิดขึ้นกับ search engine ภาษาไทยทุกแห่งคือ ปัญหาของการที่ไม่สามารถค้นหาคำที่พ้องเสียงกันแต่สะกดไม่เหมือนกัน บทความนี้ขอเสนอวิธีการค้นหาข้อมูลภาษาไทยในเครือข่ายเวิลด์ไวด์เว็บโดยใช้รหัสซาวน์เด็กซ์ช่วยในการค้นหาคำพ้องเสียงภาษาไทย ระบบนี้ได้ถูกออกแบบและพัฒนาโดยอ้างอิงตามขั้นตอนของวัฏจักรในการพัฒนาระบบ รวมทั้งพิจารณาถึงข้อจำกัดของระบบที่เกิดขึ้น เพื่อนำไปสู่แนวทางการแก้ไขปรับปรุง และเป็นแนวคิดในการประยุกต์ใช้กับงานด้านอื่นๆ ต่อไป

Title	Thai Search Engine
Student	Mr.Attapol Arjchaiyatorn
Advisor	Dr. Chotipat Pornavalai
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2002

ABSTRACT

The amount of information on the web is growing rapidly, as well as the number of new users inexperienced in the web search. Most thai people are likely to surf the web using human maintained indices or with search engine. The common problem of a thai search engine is the lack of technique to identifying words that have similar pronunciation. This project presents Thai Search Engine and technique for identifying words that have similar pronunciation. The system was well designed and developed by following Software Develoment Life Cycle (SDLC). Some limitations are found and considered as the next step of further development.

กิตติกรรมประกาศ

โครงการพัฒนาระบบสืบค้นข้อมูลภาษาไทยในอินเทอร์เน็ต จะไม่สามารถดำเนินการมาจนแล้วเสร็จได้ หากขาดความช่วยเหลือของบุคคลเหล่านี้ ผู้จัดทำจึงใคร่ขอขอบพระคุณเป็นอย่างยิ่ง

ขอขอบพระคุณ บิดา มารดา ที่ให้โอกาสและสนับสนุนทางการศึกษา

ขอขอบพระคุณ ผศ.ดร.โชติพัชร์ ภรณวลัย ผู้เป็นอาจารย์ที่ปรึกษาโครงการพัฒนาระบบงานที่กรุณาให้คำปรึกษา แนะนำในด้านต่างๆ

ผู้จัดทำหวังเป็นอย่างยิ่งว่าบทความนี้จะเป็นแนวคิดในการปฏิบัติงานเพื่อสามารถนำไปใช้ประยุกต์กับงานด้านอื่นๆ ได้เป็นอย่างดี

อรรถพล อาจไชยธร

ผู้จัดทำ

สารบัญ

สารบัญ (ต่อ)

สารบัญตาราง

หน้า

ตารางที่

สารบัญรูป

หน้า

รูปที่

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

อินเทอร์เน็ตคือแหล่งบรรจุข้อมูลจำนวนมหาศาล และเนื่องจากการที่อินเทอร์เน็ตมีอัตราการเจริญเติบโตอย่างรวดเร็ว ก็ยิ่งทำให้ปริมาณของข้อมูลภายในเครือข่ายเวิลด์ไวด์เว็บ มีขนาดเพิ่มมากขึ้นเรื่อยๆ เนื่องมาจากจำนวนของผู้ใช้ที่เพิ่มมากขึ้น ทำให้มีเว็บเพจเพิ่มมากขึ้นไปด้วย ปริมาณของข้อมูลในเว็บเพจจำนวนมากจึงเป็นอุปสรรคต่อผู้ใช้งานเครือข่ายเวิลด์ไวด์เว็บในการที่จะค้นหาข้อมูลให้ได้ตรงกับความต้องการมากที่สุด โดยเฉพาะอย่างยิ่งกับข้อมูลในเว็บเพจที่เป็นภาษาไทย แม้ว่าในปัจจุบันจะมีเว็บไซต์ให้บริการค้นหาเว็บเพจภาษาไทยอยู่มากมายหลายแห่งแล้วก็ตาม แต่เนื่องจากเว็บไซต์ให้บริการค้นหาเว็บเพจภาษาไทยส่วนมากจะอยู่ในรูปแบบของ web directory ที่มีการจัดทำข้อมูลโดยใช้คนรวบรวมและเขียนคำอธิบายของเว็บเพจ (manual indexing) ทำให้ข้อมูลที่ได้มีจำนวนจำกัด และไม่สามารถใช้ค้นหาคำหลักที่มีอยู่ในเว็บเพจได้ เนื่องจากการค้นหาข้อมูลใน web directory จะเป็นการค้นหาโดยใช้วิธีเทียบคำหลักกับคำอธิบายเว็บเพจที่จัดทำโดยบุคลากรที่เกี่ยวข้องกับ web directory นั้นๆ อย่างไรก็ดีได้มีผู้สร้างเว็บไซต์เพื่อช่วยค้นหาข้อมูลโดยใช้โปรแกรมคอมพิวเตอร์ทำหน้ารวบรวมข้อมูลเพื่อใช้ช่วยในการค้นหาโดยอัตโนมัติ (automatic indexing) เรียกเว็บไซต์ประเภทนี้ว่า search engine ตัวอย่างเว็บไซต์ประเภทนี้ในประเทศไทยเช่น www.siamguru.com, www.thaiseek.com, www.ikool.com เป็นต้น

ในปัจจุบันผลลัพธ์จากการค้นหาข้อมูลภาษาไทยในเครือข่ายอินเทอร์เน็ตมีข้อจำกัดหลายประการ หนึ่งในข้อจำกัดนั้นก็คือการที่ผู้ค้นหาข้อมูลไม่สามารถสะกดคำที่ต้องการค้นหาให้เป็นภาษาไทยได้อย่างถูกต้องหรือผู้ค้นหาสะกดคำภาษาไทยที่ต้องการจะค้นหาถูกต้องแล้ว แต่ข้อมูลที่มีอยู่ภายในเครือข่ายอินเทอร์เน็ตสะกดคำภาษาไทยคำนั้นเป็นอีกคำหนึ่งซึ่งมีการออกเสียงเหมือนกัน อันจะทำให้ไม่สามารถค้นเอกสารที่ผู้ใช้ต้องการตามคำค้นนั้นๆได้ ซึ่งสิ่งนี้เป็นปัญหาที่สำคัญที่สุดของการสืบค้นข้อมูลภาษาไทยในเครือข่ายอินเทอร์เน็ต

โดยในการพัฒนาระบบงานนี้ จะทำการแก้ไขปัญหาเรื่องการค้นหาคำภาษาไทยที่เป็นคำพ้องเสียงแต่สะกดไม่เหมือนกัน แล้วค้นหาไม่พบโดยใช้วิธีการกำหนดรหัสซาวน์เด็กซ์

1.2 วัตถุประสงค์ของการพัฒนาระบบงาน

1.2.1 เพื่อช่วยค้นหาข้อมูลในเครือข่ายอินเทอร์เน็ตโดยใช้โปรแกรมคอมพิวเตอร์ทำหน้ารวบรวมข้อมูลเพื่อใช้ช่วยในการค้นหาโดยอัตโนมัติ

1.2.2 ผู้ใช้สามารถค้นหาคำภาษาไทยซึ่งเป็นคำพ้องเสียงกันได้

1.2.3 เพิ่มประสิทธิภาพการค้นคืนเว็บเพจภาษาไทยโดยใช้เทคนิคการกำหนดรหัสซาวน์เด็กซ์

1.3 ขอบเขตของการพัฒนาระบบงาน

1.3.1 พัฒนาโปรแกรม web crawler เพื่อทำหน้าที่ค้นหาและจัดเก็บเว็บเพจภาษาไทยภายในเครือข่ายอินเทอร์เน็ต

1.3.2 พัฒนาโปรแกรม indexer เพื่อทำการประมวลผล และจัดเก็บข้อมูลภาษาไทยในเว็บเพจ โดยในการตัดคำภาษาไทยจะใช้โปรแกรม Swath ของ Nectec

1.3.3 ค้นคว้าและพัฒนาโปรแกรมเข้ารหัสซาวน์เด็กซ์ภาษาไทย เพื่อนำมาใช้ร่วมกับโปรแกรม Indexer

1.3.4 พัฒนาโปรแกรมสำหรับค้นหาข้อมูลภาษาไทย โดยทำงานผ่านเว็บบราวเซอร์

1.3.5 พัฒนาโปรแกรมสำหรับอ่านข้อมูลภายในเอกสารที่ถูกจัดเก็บอยู่ในแฟ้มข้อมูล PDF ซึ่งถูกสร้างมาจากโปรแกรม Adobe Acrobat

1.3.6 พัฒนาโปรแกรมสำหรับอ่านข้อมูลภายในเอกสารที่ถูกจัดเก็บอยู่ในแฟ้มข้อมูลที่สร้างโดยโปรแกรม Microsoft Word และ Microsoft Excel

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ช่วยอำนวยความสะดวกในการสืบค้นข้อมูลเว็บเพจภาษาไทยภายในเครือข่ายอินเทอร์เน็ต

1.4.2 ช่วยลดความผิดพลาดจากการค้นหาข้อมูลภาษาไทยที่เป็นคำพ้องเสียงกันแล้วค้นหาไม่พบ

1.4.3 เนื่องจากการค้นหาและจัดเก็บเว็บเพจทำโดยใช้โปรแกรมคอมพิวเตอร์ จึงทำให้การสืบค้นข้อมูลมีประสิทธิภาพในการค้นหาข้อมูลมากกว่าการค้นหาผ่าน web directory ซึ่งมีอยู่ตามเว็บไซท์ของไทยทั่วๆไป

1.4.4 ช่วยให้การค้นหาข้อมูลภายในเอกสารที่ถูกจัดเก็บอยู่ในรูปแบบของ Adobe Acrobat, Microsoft Word และ Microsoft Excel สามารถทำได้ง่ายขึ้น

1.5 กำหนดตารางเวลาดำเนินงาน

	ขั้นตอนการดำเนินงาน	จำนวนวัน	วันเริ่มต้น	วันสิ้นสุด
1	ศึกษาการทำงานของ search engine	31	06/08/45	05/09/45
2	ศึกษาและคัดเลือก DBMS ที่จะนำมาใช้ในการพัฒนาระบบงาน	4	06/09/45	09/09/45
3	ศึกษาการจัดเก็บข้อมูลในรูปแบบของ Vector-Space	21	10/09/45	30/09/45
4	ออกแบบฐานข้อมูลที่จะใช้กับ search engine	12	01/10/45	12/10/45
5	พัฒนาโปรแกรม web crawler และ indexer	19	16/12/45	03/01/46
6	พัฒนาโปรแกรมสำหรับค้นหาข้อมูลผ่านเว็บบราวเซอร์	6	01/01/46	06/01/46
7	ศึกษาค้นคว้าและพัฒนาโปรแกรมเข้ารหัสซาวน์เด็กซ์ภาษาไทย	98	01/10/45	06/01/46
8	ทดสอบระบบ	10	07/01/46	16/01/46

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 Hypertext Transfer Protocol (HTTP)

Hypertext Transfer Protocol คือโปรโตคอลที่ใช้สำหรับแลกเปลี่ยนข้อมูลระหว่างเว็บเซิร์ฟเวอร์และเว็บบราวเซอร์ ผ่านทางเครือข่ายอินเทอร์เน็ต การทำงานของ HTTP จะเป็นลักษณะของ การขอ request และได้ response ตอบกลับมา โดยฝ่ายไคลเอ็นต์ ซึ่งในกรณีนี้คือเว็บบราวเซอร์จะเป็นฝ่ายขอ request ไฟล์ไปยังฝ่ายเว็บเซิร์ฟเวอร์หลังจากเว็บเซิร์ฟเวอร์ได้รับ request มาแล้วก็จะทำการ response ตอบกลับไปยังเว็บบราวเซอร์ การทำงานของ HTTP จะทำงานอยู่บน TCP โดยมีหมายเลขมาตรฐานของ port ที่ใช้คือหมายเลข 80 โดยมีการทำงานของ HTTP โปรโตคอลดังรูปที่ 2.1

รูปที่ 2.1 การทำงานของ HTTP โปรโตคอล

HTTP ได้มีการกำหนดลักษณะการทำงานรวมถึงรายละเอียดต่างๆไว้ใน RFC 2068 และได้มีการพัฒนาแก้ไขเปลี่ยนแปลงใหม่อีกครั้งดังรายละเอียดใน RFC 2616 ซึ่งในปัจจุบันเวอร์ชั่นล่าสุดของ HTTP คือเวอร์ชั้น 1.1 สำหรับรายเอียดภายใน RFC สามารถหาข้อมูลเพิ่มเติมได้ที่ http://www.w3c.org/Protocols/

ใน HTTP ฝ่ายไคลเอ็นต์ เท่านั้นที่สามารถเป็นฝ่ายเริ่มต้นการติดต่อขอ request ไปยังฝ่าย server ได้โดยฝ่าย server ไม่สามารถเป็นฝ่ายที่จะขอเริ่มการติดต่อไปยังไคลเอ็นต์ ได้ไม่ว่าจะใน

กรณีใดๆก็ตาม ส่วนในการขอยุติการติดต่อสื่อสารของ HTTP นั้นสามารถกระทำได้ทั้งไคลเอ็นต์และ server ยกตัวอย่างเช่น ไคลเอ็นต์ ซึ่งในกรณีนี้คือเว็บบราวเซอร์สามารถขอยุติการติดต่อสื่อสารได้โดยการกดปุ่ม stop ที่โปรแกรมเว็บบราวเซอร์ หรือ server สามารถยุติการติดต่อสื่อสารได้โดยการ stop เว็บเซิร์ฟเวอร์

2.1.1 HTTP Requests

HTTP transaction เริ่มต้นการทำงานขึ้นโดยเว็บบราวเซอร์ขอ request ไปยังเว็บเซิร์ฟเวอร์และจะสิ้นสุด HTTP transaction เมื่อเว็บเซิร์ฟเวอร์ส่ง response ตอบกลับมายังเว็บบราวเซอร์

HTTP request ประกอบไปด้วยองค์ประกอบ 3 ส่วนด้วยกันคือ

2.1.1.1 Method – URI – Protocol/Version

ในแต่ละ HTTP request สามารถที่จะใช้ method ใดๆก็ได้ ดังที่ระบุไว้ในมาตรฐานของ HTTP ใน HTTP 1.0 method ที่สามารถใช้ได้จะมีเพียง 3 method ดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 method ของ HTTP 1.0

Method	คำอธิบาย
GET	GET เป็น method ที่ใช้บ่อยที่สุดใน HTTP หน้าที่ของ method นี้ใช้เพื่อดึงข้อมูลจาก URL ที่ระบุมาหลัง method นี้ ในกรณีที่ URL ที่ระบุมาเป็น cgi script เช่น php หรือ asp ข้อมูลที่ได้มาก็คือข้อมูลที่ได้รับมาจากการทำงานของ cgi script นั้นๆ
HEAD	method HEAD ให้ข้อมูลกลับมาคล้ายๆ method GET จะต่างกันเพียง method HEAD จะได้ผลลัพธ์กลับมาแค่ HTTP header เท่านั้น ไม่มีส่วนของ body
POST	จะคล้ายๆ method GET แต่ method POST นี้จะใช้ในการส่งข้อมูลในรูปของ block ซึ่งได้มาจาก HTML form ซึ่งจะทำให้สามารถส่งข้อมูลที่มีขนาดใหญ่มากกว่า method GET ไปยังเว็บเซิร์ฟเวอร์ได้

สำหรับ HTTP เวอร์ชั่น 1.1 จะมี method เพิ่มขึ้นมามากกว่า method ของ HTTP เวอร์ชั่น 1.0 ดังแสดงในตารางที่ 2.2

ตารางที่ 2.2 method ของ HTTP 1.1

Method	คำอธิบาย
GET	GET ใช้เพื่อดึงข้อมูลจาก URL ที่ระบุมาหลัง method นี้
HEAD	method HEAD ใช้เพื่อขอดูข้อมูลของ HTTP header
POST	method POST จะใช้ในการส่งข้อมูลในรูปของ block ซึ่งได้มาจาก HTML form ซึ่งจะทำให้สามารถส่งข้อมูลที่มีขนาดใหญ่มากกว่า method GET ไปยังเว็บเซิร์ฟเวอร์ได้
OPTIONS	method OPTIONS ใช้เพื่อสอบถามเว็บเซิร์ฟเวอร์ resource ที่เว็บเซิร์ฟเวอร์สามารถให้บริการได้
PUT	method PUT จะใช้ร่วมกับ method GET โดยจะใช้สำหรับส่งข้อมูลที่ระบุมาจาก URL ขึ้นไปยังเว็บเซิร์ฟเวอร์คล้ายๆกับคำสั่ง PUT ของ FTP โดยมากจะใช้ method PUT ในการ upload file ขึ้นไปยังเว็บเซิร์ฟเวอร์
DELETE	method DELETE ใช้สำหรับลบเอกสารออกจากเว็บเซิร์ฟเวอร์ โดยเอกสารที่จะลบจะถูกระบุอยู่ใน URI ที่ส่งมาตามหลัง method นี้
TRACE	method TRACE จะใช้ในลักษณะคล้ายๆกับคำสั่ง traceroute ของ UNIX จะมีประโยชน์มากสำหรับใช้ในการวิเคราะห์ปัญหาที่อาจจะเกิดขึ้นของการใช้งานเว็บในเครือข่ายที่ซับซ้อน

ตัวอย่างของ HTTP request แสดงได้ดังนี้

GET /http/form.php HTTP/1.1

Accept: text/plain; text/html

Accept-Language: th

Connection: Keep-Alive

Host: www.sawasdee.com

Referer: http://www.sawasdee.com/http/index.html

User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.0.3705)

Content-Length: 25

Content-Type: application/x-www-form-urlencoded

Accept-Encoding: gzip, deflate

Name=Attapol&Email=secret

จะเห็นว่า Method – URI – Protocol/version ซึ่งเป็นสิ่งที่ใช้ในการขอ request จะอยู่ในบรรทัดแรกของ request

GET /http/form.php HTTP/1.1

โดย GET คือ method ของการ request, /http/form.php คือ URI และ HTTP/1.1 คือส่วนของ Protocol/Version

URI คือส่วนที่บอกถึงตำแหน่งของ resource บนเซิร์ฟเวอร์ และจะเริ่มต้นด้วยเครื่องหมาย / สำหรับประเภทของ URI สามารถหารายละเอียดเพิ่มเติมได้ใน RFC 2396 (http://www.ietf.org/rfc/rfc2396.txt)

Protocol version คือเวอร์ชั่นของ HTTP protocol ที่ใช้ในการติดต่อกับเซิร์ฟเวอร์

2.1.1.2 Request header

Request header ประกอบด้วยข้อมูลเกี่ยวกับสภาวะแวดล้อมของไคลเอ็นต์ และ ส่วนของ body ของ request ยกตัวอย่างเช่น บราวเซอร์ของไคลเอ็นต์ว่าใช้อะไรเป็นเว็บบราวเซอร์ หรือ ภาษาที่ไคลเอ็นต์ ใช้ว่าเป็นภาษาใด และยังประกอบไปด้วยจำนวนความยาวของ entity body ที่ถูกส่งตามมาด้วยโดย request header จะถูกแบ่งออกจาก entity body โดย carriage return และ linefeed (CRLF)

2.1.1.3 Entity body

Entiry body คือส่วนที่ตามหลัง request header มาโดยมี CRLF เป็นตัวคั่นกลางระหว่าง request header กับ entity body โดยในส่วนของ entity body นี้จะใช้ในการส่งข้อมูลกลับไปยังเว็บเซิร์ฟเวอร์ ในตัวอย่างของ HTTP request ในหัวข้อที่ 2.2.1 entity body คือ Name=Attapol&Email=secret

2.1.2 HTTP Responses

HTTP response จะเป็นส่วนที่เว็บเซิร์ฟเวอร์ตอบกลับมาหลังจากได้ request จากไคลเอ็นต์ จะคล้ายๆกับ HTTP request คือ HTTP response จะประกอบไปด้วยส่วนประกอบ 3 ส่วนได้แก่

- Protocol - Status code - Description
- Response headers
- Entity body

ตัวอย่างของ HTTP Response มีดังนี้

```
HTTP/1.1 200 OK
Date: Tue, 10 Dec 2002 11:42:33 GMT
Server: Apache/1.3.27 (Unix) PHP/4.2.3
Last-Modified: Thu, 19 Dec 2002 15:53:37 GMT
ETag: "40a2d-65-3e01eb81"
Accept-Ranges: bytes
Content-Length: 101
Connection: close
Content-Type: text/html

<html>
<head>
<title>What are u looking for?</title>
</head>
<body>
Huhu ^_^
</body>
</html>
```

บรรทัดแรกของ response header จะคล้ายกับ request header คือจะบอกว่าใช้ HTTP เวอร์ชั่น 1.1, สถานะของ request สมบูรณ์ (200 คือค่าของ HTTP ที่บอกว่า success) และบอกว่าทุกอย่างทำงานได้ปกติ (OK) หลังจากนั้นก็จะตามมาด้วย response header และ

ตามมาด้วย entity body ซึ่งก็คือ HTML โดยในส่วนของ response header กับ entity body จะถูกแบ่งออกจากกันโดย carriage return/linefeed (CRLF) สำหรับค่าของ HTTP response code ที่ใช้บ่อยสามารถแสดงได้ดังตารางที่ 2.3

ตารางที่ 2.3 ค่า HTTP Response ที่ใช้บ่อย

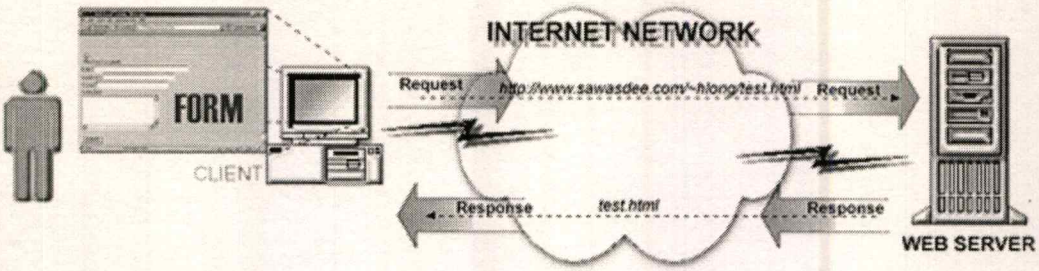
Status Code	คำอธิบาย
200	(OK)
301	(Moved Permanently) ใช้บอกให้ไคลเอ็นต์ link ไปยัง URL ใหม่
400	(Bad Request)
401	(Unauthorized)
403	(Forbidden) ในกรณีที่เซิร์ฟเวอร์ถาม user และ password ไปยังไคลเอ็นต์แล้วไคลเอ็นต์ ตอบผิดเซิร์ฟเวอร์จะตอบ response code นี้กลับมา
404	(Not Found) ใช้เมื่อเอกสารที่ request ไปใน URL ไม่ปรากฏอยู่ในเซิร์ฟเวอร์
500	(Internal Error) ถ้าเซิร์ฟเวอร์มีข้อผิดพลาดเกิดขึ้นเช่น cgi script บนเซิร์ฟเวอร์ทำงานผิดพลาด เซิร์ฟเวอร์จะตอบ response code นี้กลับมา
502	(Timed out)
503	(Service Unavailable) เมื่อเว็บเซิร์ฟเวอร์รับโหลดหนักๆและไม่สามารถประมวลผลต่อไปได้ เซิร์ฟเวอร์จะตอบ response code นี้กลับมา

2.1.3 การเชื่อมต่อของไคลเอ็นต์ และ เซิร์ฟเวอร์ ใน HTTP.

ในส่วนของเครือข่ายหรือ Network ในระบบเครือข่ายเวิร์ลไวด์เว็บ มีการเชื่อมต่อระหว่าง ไคลเอ็นต์ และเซิร์ฟเวอร์ ได้ 2 ลักษณะ คือ

2.1.3.1 การเชื่อมต่อโดยตรง

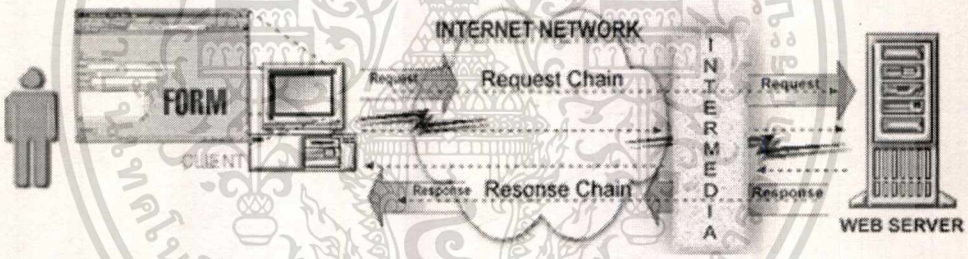
ลักษณะการทำงานคือ ไคลเอ็นต์ จะติดต่อกับเซิร์ฟเวอร์ หรือผู้ให้บริการโดยตรง ซึ่งส่วนของ ไคลเอ็นต์ จะเรียกว่า User Agent โดยมีบราวเซอร์ทำหน้าที่นี้ให้ และส่วนเซิร์ฟเวอร์ จะเรียกว่า "Origin" จะทำงานกับบราวเซอร์หรือ User Agent นี้โดยตรง ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 การเชื่อมต่อโดยตรง

2.1.3.2 การเชื่อมต่อผ่านตัวกลาง

ลักษณะการติดต่อแบบนี้ ส่วนของ User Agent ไม่สามารถติดต่อกับ Origin ได้โดยตรง นั่นคือ ต้องติดต่อผ่านตัวกลางทุกครั้งที่มีการร้องขอบริการ และการตอบสนอง ก็จะต้องผ่านตัวกลาง เช่นกัน ดังนั้น การร้องขอ หรือการตอบสนอง จะมีลักษณะเป็นเหมือนลูกโซ่ โยงผ่านเป็นช่วงๆ เรียกว่า Request Chain / Response Chain



รูปที่ 2.3 การเชื่อมต่อผ่านตัวกลาง

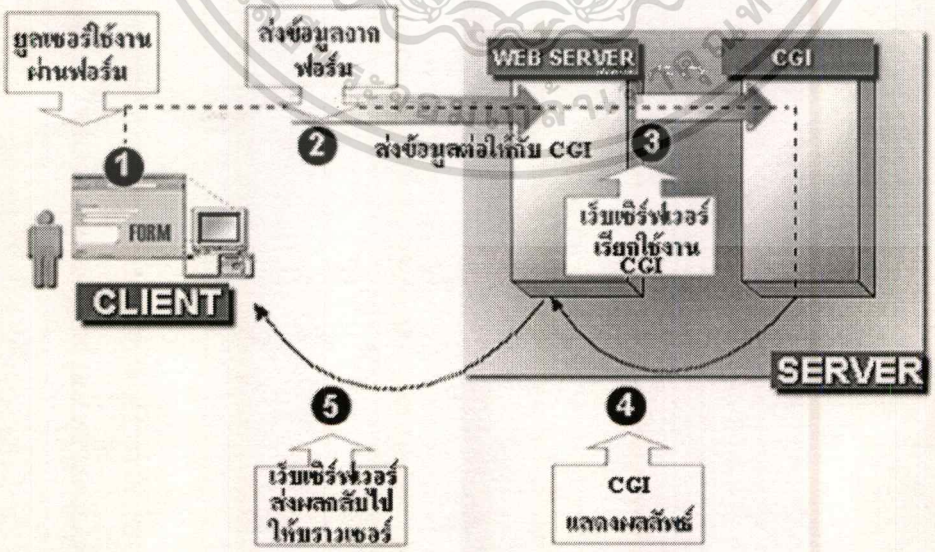
ประเภทของตัวกลาง (Intermedia) ตามข้อกำหนดของ HTTP มี 3 ประเภทคือ

- Tunnel ทำหน้าที่เชื่อมต่อเท่านั้น ตัวกลางนี้จะไม่มีหน้าที่หรืออำนาจในการเปลี่ยนข้อมูลที่วิ่งผ่าน
- Proxy ส่วนนี้สามารถปรับปรุงรายละเอียด มีการประยุกต์ใช้งานได้ ทั้ง 2 ส่วนคือ ไคลเอ็นต์ และเซิร์ฟเวอร์
- Gateway ส่วนนี้มักทำหน้าที่เชื่อมต่อ ในกรณีที่ไม่สามารถติดต่อหรือใช้งานเชื่อมต่อ กับตัว เว็บเซิร์ฟเวอร์ได้โดยตรง

ตัวกลางที่เรามักพบบ่อยในการใช้งานคือ Proxy ซึ่งมักนำมาติดตั้งเพื่อทำหน้าที่เป็น Cache Server หรือเป็น FireWall

2.2 Common Gateway Interface (CGI)

เมื่อเกิดระบบเครือข่ายเวิร์ลไวด์เว็บใช้งานจนเป็นที่นิยมดังเช่นปัจจุบัน หลายๆเว็บไซต์ เริ่มต้องการนำเสนอข้อมูลภายในองค์กร ที่เคยใช้งานกับโปรแกรมประยุกต์ของตนภายในองค์กร ผ่านเว็บเพจ หรือ โฮมเพจ (HomePage) ของตน จึงเกิดปัญหาว่าจะสามารถทำ อย่างไร ทั้งนี้เพราะทั้งสองแอพพลิเคชัน อยู่คนละส่วนกัน และวิธีการทำงานก็แตกต่างกันอย่างสิ้นเชิง ทางออกคือการพัฒนาแอพพลิเคชัน ในลักษณะเหมือนโปรแกรมประยุกต์ที่องค์กรใช้งาน อยู่ โดยอาศัยหลักการของ CGI ในการพัฒนา แต่นี่ยังเป็นเพียงแค่จุดเริ่มต้น ของความต้องการ เท่านั้น เพราะปัจจุบันเราจะเห็นว่า มีแอพพลิเคชันหลากหลายรูปแบบบนระบบเว็บ เช่น การให้บริการค้นหางาน, การให้บริการความช่วยเหลือแบบออนไลน์ เป็นต้น ซึ่งแอพพลิเคชันเหล่านี้เกิดจากความต้องการที่หลากหลาย และต่างความคิด รวมไปถึง วิสัยทัศน์ของแต่ละคนในการที่คิดประยุกต์ และร่วมสร้างกิจกรรมต่างๆ บนระบบเว็บ จนทำให้ การใช้งานระบบเว็บนี้กลายเป็นส่วนสำคัญหลักของเครือข่ายอินเตอร์เน็ตในปัจจุบันไปเสียแล้ว และด้วยความสามารถของหลักการ CGI นี้เองทำให้หลายๆ องค์กรต้องการนำมาประยุกต์ใช้ ในองค์กรจนเกิดคำที่ว่า “แอพพลิเคชันในอนาคต คือแอพพลิเคชันที่ใช้งานผ่านบราวเซอร์ หรือ ใช้งานภายใต้พื้นฐานเว็บ (Web – based หรือเรียกว่า Web – based Application)”



รูปที่ 2.4 การทำงานของ CGI

จากรูปที่ 2.4 เราจะมาให้ความหมายว่า อะไรคือ CGI จริงๆ แล้ว CGI ก็คือ หลักการหรือวิธีการของการพัฒนาแอพพลิเคชัน ที่ทำหน้าที่เสมือนประตู (Gateway) เชื่อมโยงการติดต่อ กับการทำงานอื่นๆ เพื่อให้เกิดการทำงานที่หลากหลายในการใช้งาน โดยอาศัยพื้นฐานของระบบ เว็บ หรือจะกล่าวได้ว่าทำงานควบคู่กับเว็บเซิร์ฟเวอร์ เพราะบราวเซอร์ไม่สามารถติดต่อส่วนอื่นๆ โดยตรงได้ เช่น จะติดต่อกับฐานข้อมูล เป็นต้น จำเป็นต้องติดต่อผ่านเว็บเซิร์ฟเวอร์ ไปยังส่วน ของ CGI โดยเรามักเรียกว่า "CGI โปรแกรม" หรือ "CGI แอพพลิเคชัน" หรือ "เว็บแอพพลิเคชัน" ก็ได้ ด้วยเหตุนี้เองเราจึงเห็นว่าจริงๆ แล้ว CGI แอพพลิเคชัน หรือ แอพพลิเคชัน ที่พัฒนาตาม แนวทาง CGI เป็นแอพพลิเคชันประเภท เซิร์ฟเวอร์ แอพพลิเคชัน (Server Application) หรือ แอพพลิเคชันที่ทำงานอยู่ที่ฝั่งเซิร์ฟเวอร์ โดยมีส่วนที่ทำหน้าที่ติดต่อกับผู้ขอใช้บริการ หรือ ไคลเอ็นต์ (Client) คือ เว็บเซิร์ฟเวอร์ และไคลเอ็นต์ใช้งานผ่านเว็บบราวเซอร์ ข้อดีของเซิร์ฟเวอร์ แอพพลิเคชันที่เห็นได้ชัดคือ การปรับปรุงหรือเปลี่ยนเวอร์ชันจะทำได้ง่าย โดยไม่ต้องแจกจ่าย ให้ผู้ใช้งานทุกครั้งแต่สามารถดูแลปรับปรุงได้ที่เซิร์ฟเวอร์โดยตรง พอมีวิธีการของ CGI เกิดขึ้น ปัจจุบันเราจึงได้เห็นรูปแบบของโฮมเพจที่เปลี่ยนไปจากสมัยที่เริ่มต้นของอินเทอร์เน็ตที่เคยเป็นแค่ "Static Hypermedia Document" คือเอกสารที่แสดงโดยไม่มีการเปลี่ยนแปลง ไปเป็นเอกสาร ที่สามารถเปลี่ยนแปลงรูปแบบได้ ตลอดจนเห็นเป็นโฮมเพจ ที่สามารถโต้ตอบหรือเป็น อินเตอร์แอกทีฟ (Interactive) เหมือนส่วนของอินเตอร์เฟส (interface) ของ CGI แอพพลิเคชัน ที่แปรเปลี่ยนตลอดเหมือนกับการใช้งานโปรแกรมประยุกต์นั่นเอง

2.3 Information Retrieval

การทำงานของ Search Engine จะมีการทำงานในลักษณะของ Information Retrieval โดยก่อนที่จะมีเครือข่ายอินเทอร์เน็ตเกิดขึ้น Information Retrieval เป็นเพียงแค่การทำ Index เพื่อช่วยในการค้นหาข้อมูล ตัวอย่างเช่น การค้นหารายชื่อหนังสือหรือการค้นหาชื่อผู้แต่งหนังสือในห้องสมุด แต่ในปัจจุบัน Information Retrieval มีความสลับซับซ้อนมากกว่าในสมัยก่อน เนื่องจากมีการนำเอา Information Retrieval มาใช้ในการค้นหาข้อมูลบนอินเทอร์เน็ต

การค้นหาข้อมูลอีกประเภทหนึ่งที่มีมาควบคู่กับ Information Retrieval คือ Data Retrieval ทั้งสองสิ่งนี้มีสิ่งที่แตกต่างกัน คือ ใน Data Retrieval ผลลัพธ์ที่ได้จากการค้นหาจะต้องเที่ยงตรงและแม่นยำ กล่าวคือ ควรจะต้องแสดงผลลัพธ์ออกมาให้ตรงตามที่ผู้ใช้ป้อนข้อมูลที่ต้องการค้นหา ไม่แสดงข้อมูลออกมามากเกินไปหรือน้อยเกินไป จะเห็นว่าถ้าไม่มีการเปลี่ยนแปลงของฐานข้อมูล ผลลัพธ์ที่ได้จากการค้นหาของ Data Retrieval ไม่ว่าจะเป็นการค้นหาในระยะเวลาที่ห่างกันหนึ่งเดือนหรือหนึ่งปี จะต้องให้ข้อมูลออกมาเหมือนกัน ในขณะที่ Information Retrieval จะให้ผลลัพธ์ออกมาไม่เหมือนกัน ในการค้นหาต่างช่วงระยะเวลากัน เหตุผลคือ Information Retrieval จะเป็น

การค้นหาในลักษณะของการให้ผู้ใช้งานป้อนข้อมูลที่เป็นคำพูดที่ไม่มีโครงสร้าง หรือเป็นคำพูดที่ใช้ในชีวิตประจำวัน ในขณะที่ข้อมูลที่ป้อนให้วิธีการค้นหาในแบบของ Data Retrieval จะเป็นข้อมูลที่มีรูปแบบและโครงสร้างที่แน่นอนชัดเจน ตัวอย่างเช่นข้อมูลใน Relational Database

จากที่กล่าวมาข้างต้น จะเห็นว่า Data Retrieval ไม่สามารถนำมาใช้ในการค้นหาข้อมูลที่ไม่มีรูปแบบโครงสร้างตายตัว เช่นข้อมูลในเว็บเพจบนเครือข่ายอินเทอร์เน็ตได้ แต่ Information Retrieval จะสามารถนำมาประยุกต์ใช้ในการค้นหาข้อมูลประเภทนี้ได้

ในการค้นหาข้อมูล เว็บเพจบนเครือข่ายอินเทอร์เน็ตให้ได้ข้อมูลตรงตามความต้องการของผู้ใช้งานมากที่สุด Information Retrieval จะต้องสามารถแปลงข้อมูลที่ผู้ใช้งานต้องการค้นหา ให้อยู่ในรูปแบบที่สามารถนำไปใช้ค้นหาข้อมูลได้ถูกต้องตรงกับความต้องการของผู้ใช้งานมากที่สุด เป้าหมายหลักของ Information Retrieval คือ เพื่อค้นคืนเอกสารทั้งหมดที่ตรงกับคำถามที่ผู้ค้นหาป้อนข้อมูลลงไป โดยจะต้องดึงเอกสารที่ไม่เกี่ยวข้องออกมาให้น้อยที่สุด หรือไม่ดึงเอกสารที่ไม่มีความเกี่ยวข้องกันออกมาเลย

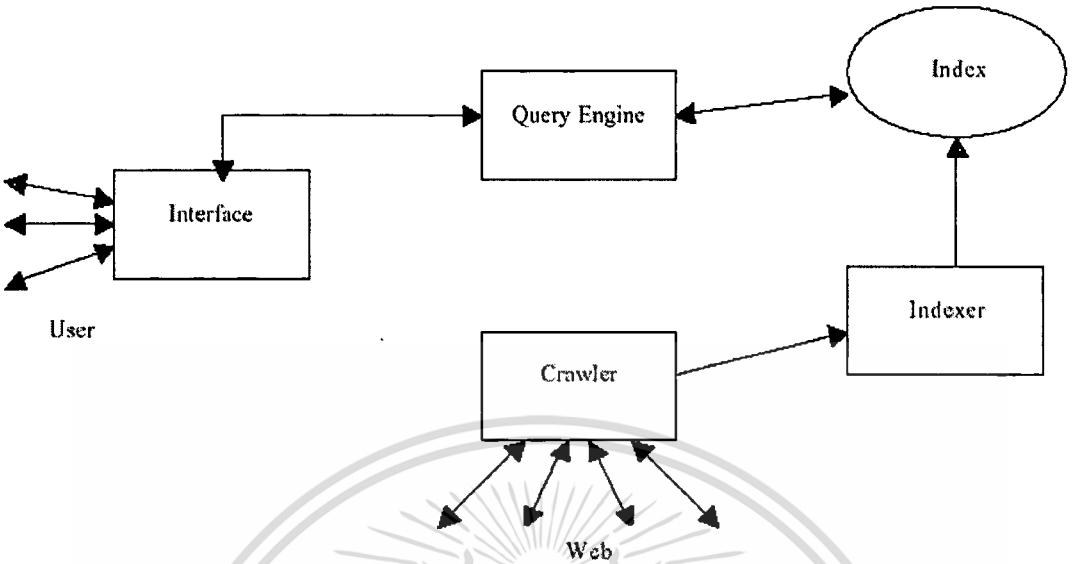
การนำ Information Retrieval สำหรับนำไปใช้ในการค้นหาข้อมูลในเว็บเพจบนเครือข่ายอินเทอร์เน็ตจะมีความแตกต่างจาก Information Retrieval ในสมัยก่อนหน้าที่จะมีเครือข่ายอินเทอร์เน็ต กล่าวคือ ในสมัยแรกๆ Information Retrieval จะสามารถเข้าถึงข้อมูลในเอกสารนั้นๆทั้งฉบับ ในขณะที่การนำ Information Retrieval ไปใช้ใน Search Engine ในสมัยปัจจุบันจะไม่สามารถทำเช่นนั้นได้ เพราะระบบ Search Engine ไม่สามารถที่จะจัดเก็บข้อมูลของเอกสารทั้งฉบับ ลงในตัวระบบได้ทั้งหมด เนื่องจากข้อจำกัดทางด้านทรัพยากรของระบบ Search Engine เอง และปริมาณของข้อมูลของหน้าเว็บเพจที่สามารถเพิ่มขึ้นได้อย่างไม่มีขีดจำกัดภายในเครือข่ายอินเทอร์เน็ต

2.4 การทำงานของ Search Engine

Search Engine ของต่างประเทศในปัจจุบัน โดยมากมักจะตั้งอยู่ในประเทศสหรัฐอเมริกา และในบรรดา search engine ชื่อดังเหล่านี้โดยมากมักจะปกปิดสถาปัตยกรรมและการทำงานของตัวเองไว้เป็นความลับ อย่างไรก็ตามมี search engine ดังๆบางแห่งเผยแพร่ข้อมูลเหล่านี้ออกมา ไม่ปกปิดเป็นความลับ เช่น Altavista และ Google โดยทั้งสองแห่งมีวิธีการทำงานดังนี้

2.4.1 Altavista

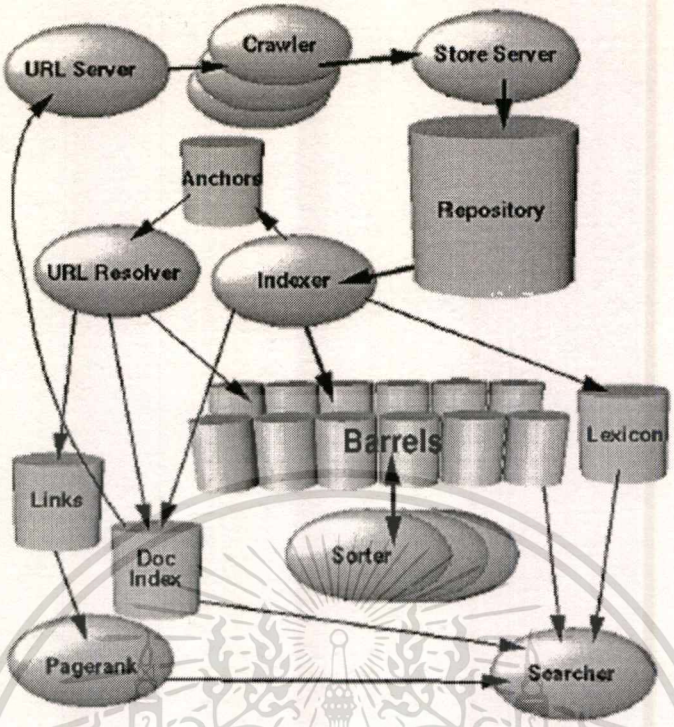
การทำงานของ Altavista จะสามารถแบ่งออกได้เป็น 2 ส่วนหลักๆด้วยกันคือส่วนแรกเป็นส่วนของ user interface และ query engine สำหรับรับข้อมูลและค้นหาข้อมูลให้กับผู้ใช้ ส่วนที่สองคือ crawler และ indexer สำหรับท่องไปเก็บของมูลบนอินเทอร์เน็ตมาจัดเก็บลงในฐานข้อมูล โดยมีการทำงานดังรูปที่ 2.5



รูปที่ 2.5 สถาปัตยกรรมของ Altavista

2.4.2 Google

Google มีการทำงานดังแสดงในรูปที่ 2.6 โดยจะเริ่มจาก URL Server จะส่ง list ของ URL ออกไปให้ crawler ไปดึงหน้าเว็บเพจส่งให้ Store Server เพื่อให้ Store Server compress หน้าเว็บเพจเหล่านั้นและจัดเก็บลงใน Repository หลังจากนั้น Indexer จะมาดึงข้อมูลจาก Repository มาทำการประมวลผลก่อนจัดเก็บลงใน Barrel แล้วในส่วนของการค้นหาเมื่อมีผู้ใช้ป้อน query มาก็จะถูก searcher ไปค้นข้อมูลมาให้โดยข้อมูลที่ค้นหาเจอจะผ่านการจัดลำดับโดย Pagerank ก่อนส่งผลลัพธ์กลับไปให้ผู้ใช้งาน



รูปที่ 2.6 สถาปัตยกรรมของ Google

2.5 Search Engine Robots

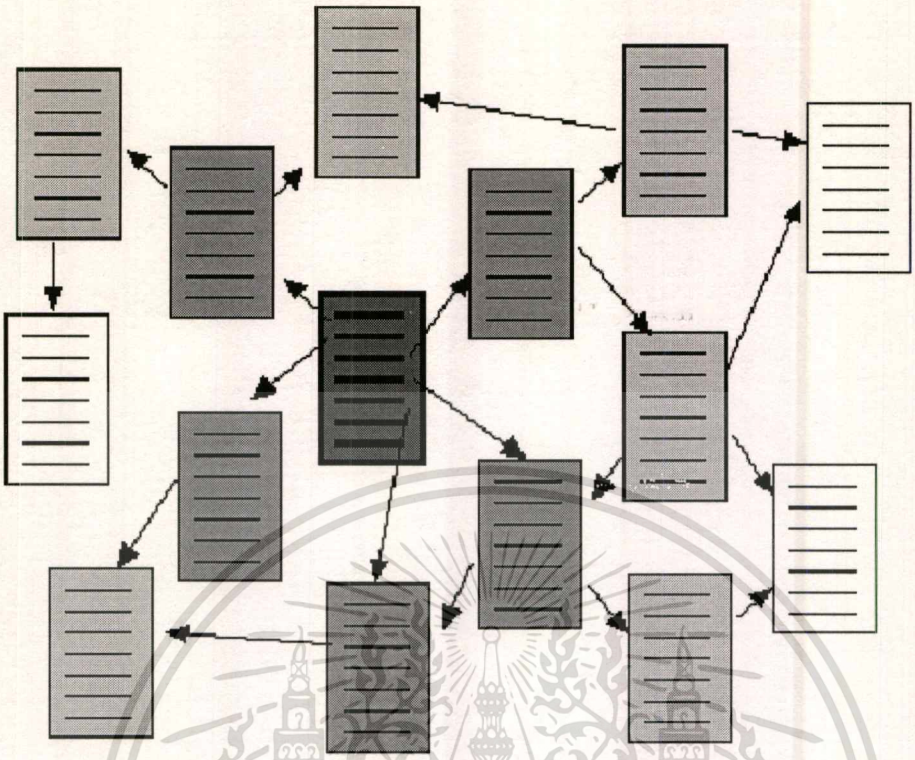
Search Engine Robot หรือ Web Crawler คือ โปรแกรมที่ทำหน้าที่ท่องไปตามเว็บไซต์เก็บเอาข้อมูล และลิงค์ที่เจออยู่ภายในหน้าเว็บเพจกลับมาทำเป็นอินเด็กซ์สำหรับใช้ช่วยในการค้นหาข้อมูลใน Search Engine

Search Engine ที่ได้รับความนิยมดัง ๆ ในต่างประเทศเช่น Google, Altavista, Hotbot หรือ All The Web ล้วนแล้วแต่ใช้ Web Crawler เป็นตัวรวบรวมข้อมูลมาใส่ยังเว็บไซต์ของตัวเอง โดยลิงค์ที่ Web Crawler อ่านเจอทุกลิงค์จะถูก Web Crawler นำเอาไปเป็นอินพุทสำหรับท่องไปยังเว็บไซต์เหล่านั้นต่อไป

วิธีที่ Web Crawler ใช้ในการเดินทางท่องไปตามเว็บไซต์สามารถแบ่งออกได้ดังต่อไปนี้

2.5.1 Breadth-First Crawling

วิธีนี้เริ่มต้นการทำงานโดย Web Crawler จะได้รับ URL เริ่มต้นมา 1 URL หรือมากกว่านั้น เพื่อใช้เป็นจุดเริ่มต้นในการเดินทางท่องไปตามเว็บไซต์เพื่อเก็บรวบรวมข้อมูล หลังจาก Web Crawler ท่องไปยังเว็บไซต์เหล่านั้นแล้วก็จะค้นพบ URL ในเว็บเหล่านั้นอีก และก็จะนำ URL ทั้งหมดที่ค้นพบในเว็บไซต์เหล่านั้นกลับมาเก็บยังฐานข้อมูลของ Web Crawler ก่อนจะเดินทางไปยัง URL ที่เจอเหล่านั้นต่อไปตามลำดับดังแสดงได้ในรูปที่ 2.7

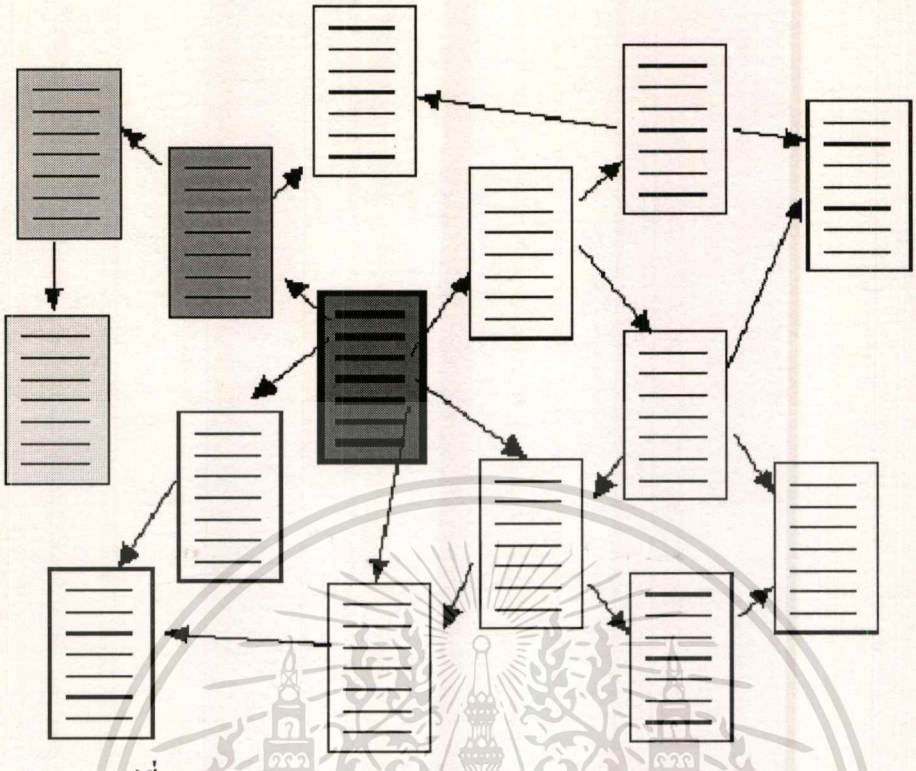


รูปที่ 2.7 การทำงานของ Web Crawler แบบ Breadth-First Crawling

จากรูปที่ 2.7 URL เริ่มต้นจะถูกแสดงด้วยสีที่เข้มที่สุด และ URL ลำดับถัดไปที่ Web Crawler จะท่องต่อไปจะถูกแสดงด้วยสีที่เข้มน้อยลงมาตามลำดับจนถึง URL สุดท้ายจะถูกแสดงด้วยสีขาวดังรูป

2.5.2 Depth-First Crawling

วิธีนี้เริ่มต้นการทำงานโดย Web Crawler จะได้รับ URL เริ่มต้นมา 1 URL หรือมากกว่านั้นเพื่อใช้เป็นจุดเริ่มต้นในการเดินทางออกไปเก็บข้อมูล และ URL ลำดับแรกในเพจต่อๆ ไปที่ถูก Web Crawler ค้นพบ จะถูกนำมาเป็นอินพุทสำหรับ Web Crawler ในการเดินทางออกไปเก็บข้อมูลต่อไป ดังแสดงได้ในรูปที่ 2.8



รูปที่ 2.8 การทำงานของ Web Crawler แบบ Depth-First Crawling

จากรูปที่ 2.8 URL แรกที่ Web Crawler ใช้เป็นจุดเริ่มต้นจะถูกแสดงด้วยสีที่เข้มที่สุด และ URL ถัดไปที่ Web Crawler จะท่องไปจะถูกแสดงด้วยสีที่เข้มน้อยลงมาตามลำดับ จนถึง URL สุดท้ายจะถูกแสดงด้วยสีขาวดังรูป

2.6 Robots Exclusion Protocol

การทำงานของ crawler จะต้องเป็นไปตาม robots exclusion standard ซึ่งเป็นมาตรฐานที่มีการเผยแพร่ครั้งแรกเมื่อปี ค.ศ.1994 เพื่อใช้เป็นมาตรฐานในการป้องกันไม่ให้ crawler, robot หรือ spider เข้าไปอ่านเว็บเพจในบริเวณที่เจ้าของเว็บไซต์ไม่ต้องการ

ในการเดินทางไปเก็บข้อมูลตามเว็บไซต์ Robot ควรจะต้องตรวจสอบหาไฟล์ชื่อ Robots.txt ที่ Root Directory ของเว็บไซต์นั้น ๆ ซึ่งไฟล์ Robots.txt นี้จะเป็น Text File ที่ถูกกำหนดขึ้น โดยมีรูปแบบของไฟล์ Robots.txt ดังนี้คือ ไฟล์ robots.txt จะประกอบด้วย record ของคำสั่ง 1 คำสั่งหรือมากกว่านั้น โดยในแต่ละ record จะถูกแบ่งโดย การขึ้นบรรทัดใหม่ หรือ บรรทัดว่างๆ (แบ่งโดย CR, CR/New Line หรือ New Line) และใน record แต่ละบรรทัดจะมีรูปแบบ "<field>:<optionalspace><value><optionalspace>" โดยชื่อของ field จะไม่สนใจว่าเป็นตัวอักษรตัวเล็กหรือตัวใหญ่

เราสามารถเขียน comment ใน robots.txt โดยใช้ '#' ซึ่งเป็น comment ที่ใช้ใน UNIX bourne shell ทั่วๆไป และ record จะเริ่มต้นด้วยคำสั่ง User-agent จำนวน 1 บรรทัดหรือมากกว่านั้น ตามด้วยคำสั่ง Disallow คำสั่งที่มีรูปแบบนอกเหนือไปจากนี้ให้ crawler ไม่ต้องสนใจและสามารถ ignore ได้ทันที

รูปแบบการใช้คำสั่งใน robots.txt จะมีลักษณะดังสามารถอธิบายได้ดังต่อไปนี้

- User-agent

  ค่าที่ตามหลังคำสั่งนี้คือชื่อของ crawler ที่ record นี้มีผลบังคับให้ใช้เป็น access policy ถ้ามี User-agent หลายบรรทัดก็หมายถึง เป็น record ที่มีผลบังคับเป็น access policy ของ crawler มากกว่า 1 ตัว และถ้าค่าที่ตามหลัง User-agent เป็นเครื่องหมาย '*' หมายถึงเป็น record ที่มีผลบังคับให้เป็น access policy ของ crawler ทุกๆตัว

- Disallow

  ค่าที่ตามหลังคำสั่งนี้คือ URL ที่ห้ามไม่ให้ crawler เข้าไปอ่านข้อมูล โดย URL นี้อาจจะเป็นเพียงแค่ path เช่น /support/ หรือเป็น URL แบบประกอบด้วยชื่อไฟล์เช่น /support/index.html ซึ่งถ้ากำหนดค่าเป็น path เช่น /support/ จะมีผลทำห้ามไม่ให้ crawler เข้าไปอ่านเว็บเพจทุกไฟล์ใน /support/ ในขณะที่การกำหนดแบบใส่ชื่อไฟล์เช่น /support/index.html จะเป็นการห้ามไม่ให้ crawler อ่าน index.html เพียงไฟล์เดียวเท่านั้น

ตัวอย่างการเขียนไฟล์ robots.txt ที่เขียนขึ้นมาเพื่อห้ามไม่ให้ crawler ทุกตัวเข้าไปอ่านข้อมูลเว็บเพจทุกไฟล์ภายใน path /cyberworld/map/ และ /tmp/ และห้ามไม่ให้อ่านเว็บเพจ /foo.html มีรูปแบบดังแสดงได้ดังนี้

```
# robots.txt for
# http://www.example.com/
User-agent: *
Disallow: /cyberworld/map/
Disallow: /tmp/
Disallow: /foo.html
```

นอกเหนือจากไฟล์ Robots.txt แล้ว เจ้าของเว็บไซต์ยังสามารถกำหนดหน้าเว็บเพจเพียงบางหน้า ที่ไม่ต้องการให้ Robot วิ่งเข้ามาเก็บข้อมูลได้โดยอาศัยกลไกของ Robots META Tag รูปแบบของ Robots META Tag จะถูกใส่อยู่ในส่วนของ HTML <HEAD></HEAD> ดังแสดงได้ดังนี้

<meta name="ROBOTS" content="NOINDEX">

<meta name="ROBOTS" content="NOFOLLOW">

<meta name="ROBOTS" content="NOINDEX,NOFOLLOW">

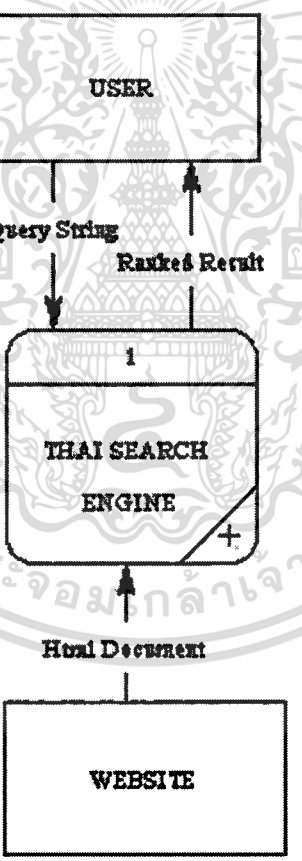
<meta name="ROBOTS" content="INDEX,FOLLOW">

บทที่ 3

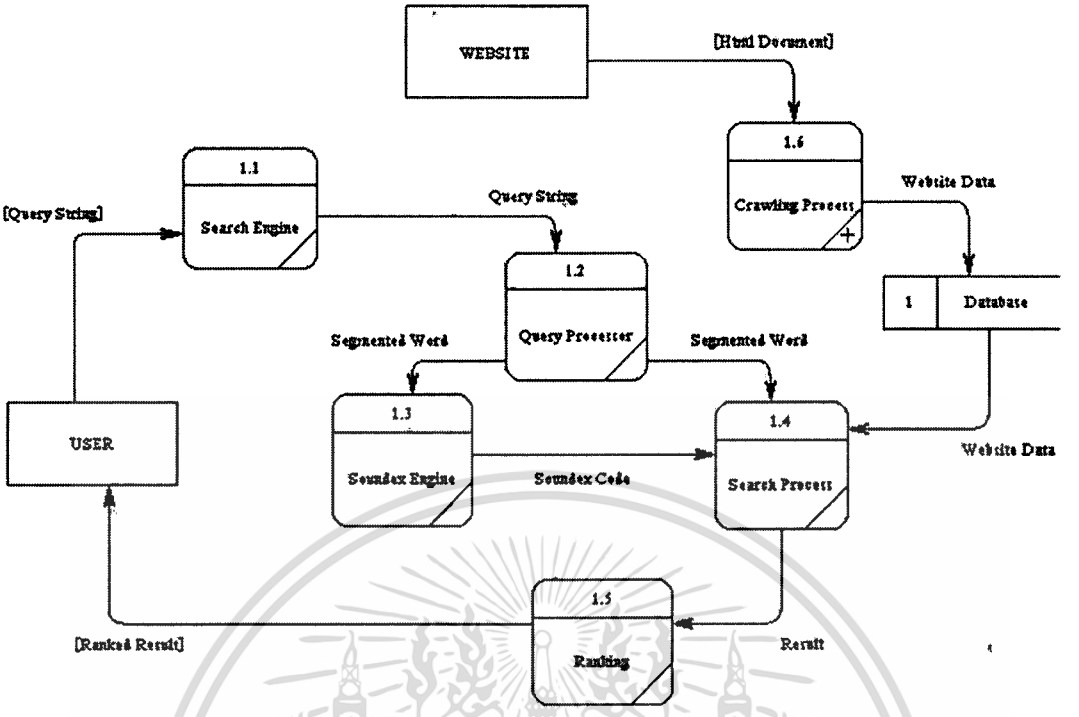
การออกแบบระบบ

3.1 วิเคราะห์ระบบสืบค้นข้อมูลภาษาไทย

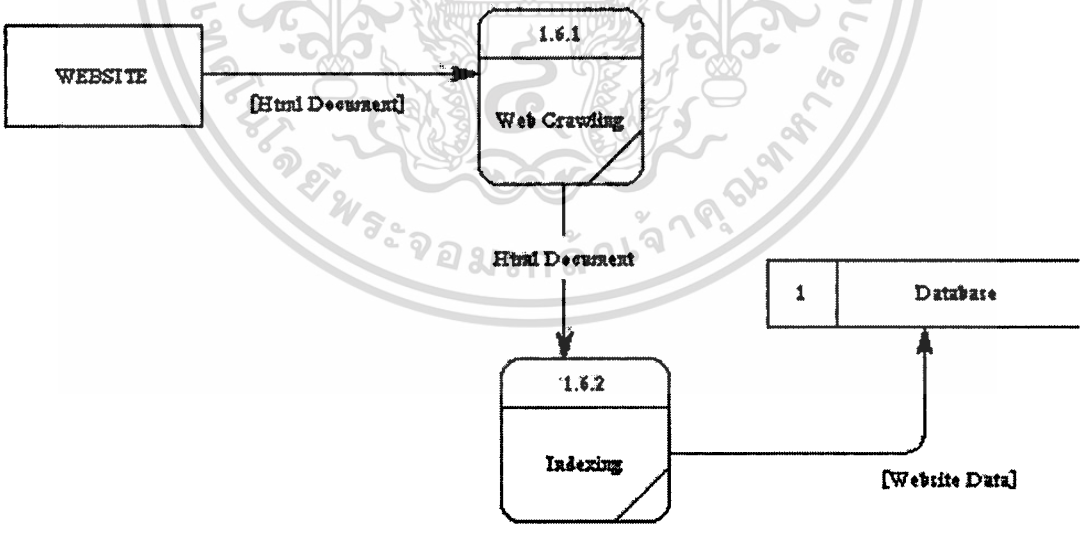
หลังจากที่ได้วิเคราะห์ระบบ Search Engine แล้วสามารถออกแบบระบบได้ด้วยแผนภาพต่างๆ ซึ่งประกอบด้วย context diagram และ data flow diagram โดยมีรายละเอียดดังต่อไปนี้



รูปที่ 3.1 แผนภาพคอนเท็กซ์ไดอะแกรมของระบบ Search Engine



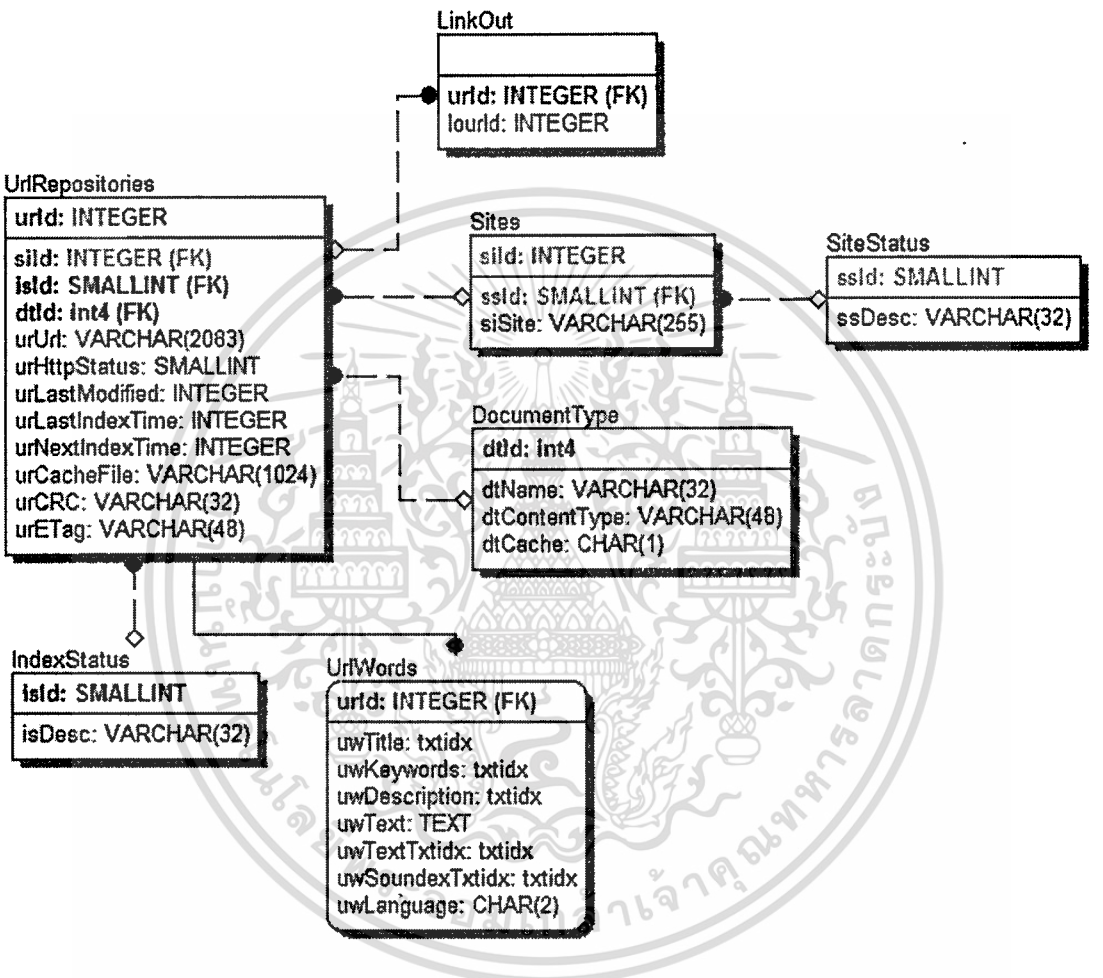
รูปที่ 3.2 Data Flow Diagram Level 1



รูปที่ 3.3 Data Flow Diagram Level 2

3.2 การออกแบบระบบฐานข้อมูล

การออกแบบฐานข้อมูลของระบบ Search Engine จะพิจารณาจากข้อมูลต่างๆที่ได้ศึกษารายละเอียดการทำงานของระบบ จากนั้นจะทำการสร้างความสัมพันธ์ระหว่างตารางต่างๆ ซึ่งสามารถแสดงได้ด้วย Entity Relationship Diagram (E-R Diagram) ดังรูปที่ 3.4



รูปที่ 3.4 Entity – Relationship Diagram

3.3 Data Dictionary

จากการวิเคราะห์และออกแบบระบบฐานข้อมูลโดยวิธีใช้ Entity Relational Model สามารถนำไปสร้างเป็นตารางข้อมูล ซึ่งมีตารางข้อมูลที่ใช้ในระบบดังนี้

ตารางที่ 3.1 ตารางเก็บรายละเอียดของ URL

ชื่อตาราง URLREPOSITORIES

ความหมาย เก็บรายละเอียดของ URL

ชื่อเขตข้อมูล	ความหมาย	ชนิดข้อมูล	คีย์	อ้างอิงจากตาราง
URID	รหัส URL	INTEGER	PK	
SIID	รหัสของเว็บไซต์	INTEGER	FK	SITES
ISID	รหัสสถานะการจัดเก็บ	SMALLINT	FK	INDEXSTATUS
DTID	รหัสประเภทเอกสาร	INTEGER	FK	DOCUMENTTYPE
URURL	URL ของเว็บเพจ	VARCHAR(2083)		
URHTTPSTATUS	HTTP Code ของ URL	SMALLINT		
URLASTMODIFIED	วันที่ URL มีการเปลี่ยนแปลงครั้งล่าสุด	INTEGER		
URLASTINDEXTIME	วันที่จัดเก็บ URL	INTEGER		
URNEXTINDEXTIME	วันที่ต้องตรวจสอบ URL อีกครั้ง	INTEGER		
URCACHEFILE	เก็บ PATH ของ cache file ของเว็บเพจ	VARCHAR(1024)		
URCRC	เก็บค่า CRC ของเว็บเพจ	VARCHAR(32)		
URETAG	เก็บค่า ETAG ของเว็บเพจ	VARCHAR(48)		

ตารางที่ 3.2 ตารางเก็บข้อมูลเว็บไซต์

ชื่อตาราง SITES

ความหมาย เก็บรายชื่อ domain name ของเว็บไซต์

ชื่อเขตข้อมูล	ความหมาย	ชนิดข้อมูล	คีย์	อ้างอิงจากตาราง
SIID	รหัสของเว็บไซต์	INTEGER	PK	
SSID	สถานะของเว็บไซต์	SMALLINT	FK	SITESTATUS
SISITE	ชื่อ domain name	VARCHAR(255)		

ตารางที่ 3.3 ตารางสถานะของเว็บไซต์

ชื่อตาราง SITESTATUS

ความหมาย เก็บสถานะของเว็บไซต์

ชื่อเขตข้อมูล	ความหมาย	ชนิดข้อมูล	คีย์	อ้างอิงจากตาราง
SSID	สถานะของเว็บไซต์	SMALLINT	PK	
SSDESC	คำอธิบายสถานะ	VARCHAR(32)		

ตารางที่ 3.4 ตารางสถานะการจัดเก็บเว็บเพจ

ชื่อตาราง INDEXSTATUS

ความหมาย เก็บสถานะการจัดเก็บเว็บเพจ

ชื่อเขตข้อมูล	ความหมาย	ชนิดข้อมูล	คีย์	อ้างอิงจากตาราง
ISID	รหัสสถานะการจัดเก็บ	SMALLINT	PK	
ISDESC	คำอธิบายสถานะ	VARCHAR(32)		

ตารางที่ 3.5 ตารางเก็บประเภทของเอกสาร

ชื่อตาราง DOCUMENTTYPE

ความหมาย เก็บรายชื่อประเภทของเอกสาร

ชื่อเขตข้อมูล	ความหมาย	ชนิดข้อมูล	คีย์	อ้างอิงจากตาราง
DTID	รหัสประเภทเอกสาร	INTEGER	PK	
DTNAME	ชื่อประเภทเอกสาร	VARCHAR(32)		
DTCONTENTTYPE	MIME TYPE	VARCHAR(48)		
DTCACHE	เก็บลง cache?	CHAR(1)		

ตารางที่ 3.6 ตารางการลิงค์ของเว็บเพจ

ชื่อตาราง LINKOUT

ความหมาย เก็บการลิงค์ของเว็บเพจ

ชื่อเขตข้อมูล	ความหมาย	ชนิดข้อมูล	คีย์	อ้างอิงจากตาราง
URID	รหัส URL	INTEGER	FK	URLREPOSITORIES
LOURID	รหัส URL ที่ถูกลิงค์	INTEGER		

ตารางที่ 3.7 ตารางเก็บข้อมูลภายในเว็บเพจ

ชื่อตาราง URLWORDS

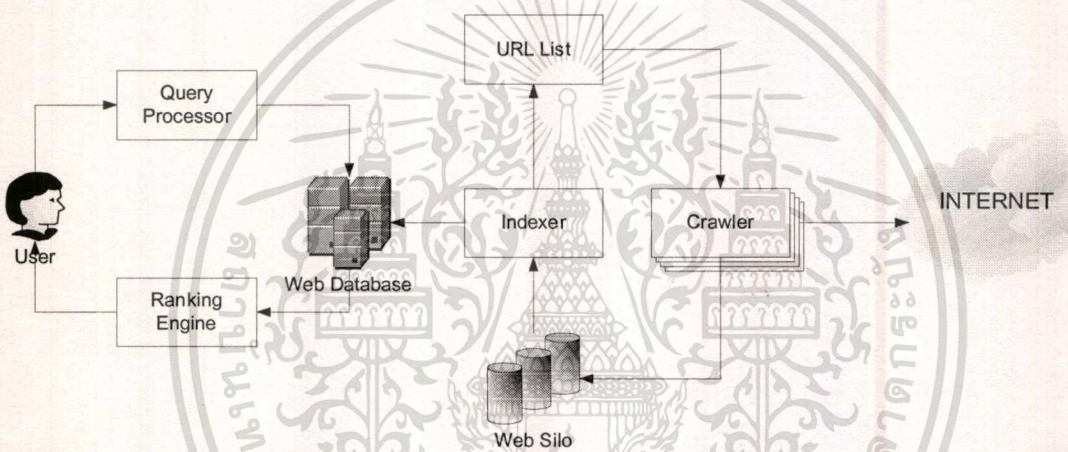
ความหมาย เก็บข้อมูลภายในเว็บเพจ

ชื่อเขตข้อมูล	ความหมาย	ชนิดข้อมูล	คีย์	อ้างอิงจากตาราง
URID	รหัส URL	INTEGER	FK	URLREPOSITORIES
UWTITLE	TITLE ของเว็บเพจ	TXTIDX		
UWKEYWORDS	META KEYWORD	TXTIDX		
UWDESCRIPTION	META DESCRIPTION	TXTIDX		
UWTEXTTXTIDX	รายละเอียดของเว็บเพจ	TXTIDX		
UWSOUNDEXTXTIDX	soundex รายละเอียดของเว็บเพจ	TXTIDX		
UWLANGUAGE	ภาษาที่ใช้ในเว็บเพจ	CHAR(2)		

บทที่ 4
การทำงานของโปรแกรมสืบค้นข้อมูลภาษาไทย

4.1 โครงสร้างของโปรแกรมสืบค้นข้อมูลภาษาไทย

จากการวิเคราะห์ และออกแบบระบบสืบค้นข้อมูลภาษาไทยในอินเทอร์เน็ต ได้แยกการทำงานของระบบออกเป็นส่วนๆ ดังแสดงในรูปที่ 4.1



รูปที่ 4.1 โครงสร้างของโปรแกรมสืบค้นข้อมูลภาษาไทย

ในแต่ละส่วนของระบบที่ทำงานร่วมกัน จะสามารถแบ่งการทำงานของระบบออกเป็นส่วนหลักๆได้ 2 ส่วนด้วยกันคือ

4.1.1 ส่วนจัดเตรียมเอกสารสำหรับการค้นหา

การทำงานของส่วนนี้มีหน้าที่คอยจัดเตรียมเอกสารสำหรับการค้นหา, ค้นหาที่อยู่ของเว็บเพจโดยอัตโนมัติ และแยก link ออกจากหน้าเว็บเพจ โดยส่วนจัดเตรียมเอกสารสำหรับการค้นหาประกอบด้วยส่วนต่างๆดังนี้

- Crawler
- Indexer
- URL List
- Web Database

- Web Silo

การทำงานของส่วนจัดเตรียมเอกสารจะเริ่มจาก Crawler จะดึงเอา URL ของหน้าเว็บเพจที่จะทำการ index ออกมาจาก URL List แล้ววิ่งไปยัง URL นั้นๆเพื่อนำเอาเว็บเพจมา compress และ save เก็บไว้ใน web silo หลังจากนั้น indexer ก็จะดึงเอาหน้าเว็บเพจออกมาจาก web silo ทำการ uncompress หน้าเว็บเพจนั้นๆก่อนจะทำการประมวลผล ในขั้นตอนการประมวลผลข้อความในเว็บเพจจะถูกแบ่งย่อยออกเป็นคำๆ ถ้าเป็นภาษาอังกฤษก็จะสามารถตัดแบ่งคำได้โดยพิจารณาจากช่องว่างหรือตัวอักษรพิเศษเช่นเครื่องหมาย comma หรือ dash หลังจากนั้นก็จะทำการ stemming คำที่ได้มาจากการตัดแบ่ง ส่วนเอกสารที่เป็นภาษาไทยก็จะตัดคำโดยใช้โปรแกรม SWATH หลังจากตัดคำเสร็จแล้ว indexer จะทำการจัดเก็บ index ของคำพร้อมน้ำหนักความสำคัญของคำนั้นๆลงใน web database โดยในขั้นตอนของการประมวลผลนี้ตัว indexer จะทำการจัดเก็บ link ที่พบในเอกสารพร้อมกับ link structure ลงใน URL list ด้วย

4.1.2 ส่วนติดต่อกับผู้ใช้โปรแกรมสืบค้นข้อมูลภาษาไทย

การทำงานของส่วนนี้จะเป็นส่วนที่ทำหน้าที่ติดต่อกับผู้ใช้งาน search engine ประกอบได้ด้วยส่วนต่างๆดังนี้

- Query Processor
- Ranking Engine

การทำงานในส่วนนี้จะเริ่มจากผู้ใช้งานระบบสืบค้นข้อมูล ป้อนคำที่ต้องการจะค้นหา หลังจากนั้น query processor จะทำการตัดแบ่งคำที่ผู้ใช้ป้อนมาออกเป็นคำย่อยๆ ในกรณีที่คำที่ป้อนมาเป็นภาษาอังกฤษ query processor ก็จะทำการ stemming คำๆนั้นด้วย หลังจากนั้น คำที่ผ่านการประมวลผลจาก query processor ก็จะถูกส่งไปค้นหาเอกสารที่มีคำๆนั้นใน web database ในขั้นตอนนี้จะได้เอกสารที่มีคำที่ผู้ใช้ต้องการค้นหาออกมา ตัว Ranking Engine จะนำเอาเอกสารเหล่านั้นไปจัดลำดับความสำคัญของผลลัพธ์ ก่อนจะแสดงผลลัพธ์จากการจัดเรียงกลับไปยังผู้ใช้เป็นอันสิ้นสุดขั้นตอนการค้นหาข้อมูล

4.2 การทำงานของส่วนค้นหาคำพ้องเสียงโดยใช้ซาวน์เด็กซ์

ซาวน์เด็กซ์คือวิธีของการสร้างรหัสสำหรับค้นหาข้อมูล โดยคำนึงถึงการออกเสียงของคำนั้นๆเป็นหลัก กล่าวคือคำใดที่มีการออกเสียงที่เหมือนกันก็ควรจะให้รหัสซาวด์เด็กซ์ที่เหมือนกันออกมาด้วย เพื่อที่ว่าในการค้นหาข้อมูลจะได้นำเอารหัสซาวน์เด็กซ์ไปทำการค้นหา เพื่อช่วยลดความผิดพลาดในการค้นหาคำที่พ้องเสียงกันเช่นคำว่า 'บรรได' และ 'บันได'

การสร้างซาวน์เด็กซ์ในบทความนี้จะมีวิธีการกำหนดรหัสของซาวน์เด็กซ์ โดยจะมีวิธีการแปลงคำภาษาไทยให้เป็นรหัสซาวน์เด็กซ์ ซึ่งจะมีวิธีการสร้างรหัสซาวน์เด็กซ์โดยจะมองว่าคำทุกคำประกอบไปด้วย กลุ่มของพยัญชนะต้น ตามด้วยกลุ่มของสระ และกลุ่มของพยัญชนะที่เป็นตัวสะกด โดยการสร้างรหัสซาวน์เด็กซ์กลุ่มของพยัญชนะต้น จะแบ่งกลุ่มพยัญชนะต้นออกเป็นกลุ่มๆ ตามลักษณะการออกเสียงที่คล้ายกัน โดยจะโดยใช้วิธีของ Tassawut Duangpanyasawang and Boonserm Kijsirikul มาปรับปรุงเปลี่ยนแปลงให้ดีขึ้นได้ผลออกมาดังตารางที่ 4.1

ตารางที่ 4.1 รหัสซาวน์เด็กซ์ของกลุ่มพยัญชนะต้น

พยัญชนะต้น	รหัสซาวน์เด็กซ์
ก กล กร กว	ก
ข ค ฆ คล คร ควฃ ขร ขล ฆ ฅ	ค
ง หง	ง
จ จร	จ
ช ชร ฌ ฉ	ช
ส ซ ษ ศ ศร ศล สร สล ทร ซร	ส
ย อย หญ ญ หย	ย
ด ดร ฎ	ด
ต ตร ตล ฏ	ต
ท ทร ฐ ถ ฑ ฒ ธ	ท
น ณ หน	น
บ บร บล	บ
ป ปร ปล	ป
พ พร พล ผ ผล ภ	พ
ฝ ฝล ฝร ฟ ฟล ฟร	ฟ
ม หม	ม

พยัญชนะต้น	รหัสซาวน์เด็กซ์
ล ร ฬ หล หร ฤ ฦ	ล
ว หว	ว
อ	อ
ห ฮ	ฮ

และมีการกำหนดรหัสซาวน์เด็กซ์ของกลุ่มของสระได้ดังตารางที่ 4.2 ดังนี้

ตารางที่ 4.2 รหัสซาวน์เด็กซ์ของกลุ่มสระ

กลุ่มของสระ	รหัสซาวน์เด็กซ์
-ะ	A
-ไม้หันอากาศ(แม่กน) –รร –รร(แม่กน)	B
-อัม -ำ –รรม	C
-อัว -อัวะ	D
-อิ -อี –ฤ-	E
-อึ –อือ	F
-อุ –อู	G
ไ- ใ- ไ-ย ไ-ร ไ-ล –ัย	H
เ-า –าว	I
เ-ะ	J
เ-	K
แ-ะ	L
แ-	M
โ-ะ	N
โ-	O
เ-าะ	P
-อ	Q
เ-อะ	R
เ-อ	S
เ-	T

กลุ่มสระ	รหัสซาวน์เด็กซ์
เ-ียะ	U
เ-ีย	V
เ-อะ	W
เ-อ	X
-ไม้หันอากาศ(แม่กก) –รร(แม่กก)	Y
-ไม้หันอากาศ(แม่กด) –รร(แม่กด)	Z
–า	1
-ไม้หันอากาศ(แม่กบ) –รร(แม่กบ)	2
-ไม้หันอากาศ(แม่กง)	3
-ว-	4

โดยที่ – ในตารางหมายถึงพยัญชนะต้น และมีกฎอยู่ว่า ไม้เอก, ไม้โท, ไม้ตรี และไม้จัตวา จะโดนตัดทิ้ง เราจะไม่นำมาสร้างรหัสซาวน์เด็กซ์ และเราจะตัดพยัญชนะในตำแหน่งที่มีตัวการันต์อยู่ข้างบน ทิ้งไม่นำมาสร้างรหัสซาวน์เด็กซ์เช่นกัน เช่นคำว่า 'ศักดิ์' เราจะตัด 'ดิ์' ทิ้ง โดยไม่นำพยัญชนะในตำแหน่งนี้ทั้งหมดมาสร้างรหัสซาวน์เด็กซ์ และในกรณีที่คำๆนั้นมีกลุ่มสระก่อนหน้าพยัญชนะต้น ให้เลื่อนพยัญชนะต้นมาอยู่หน้ากลุ่มสระและจึงทำการสร้างรหัสซาวน์เด็กซ์

สุดท้ายคือตารางรหัสซาวน์เด็กซ์ของพยัญชนะที่เป็นตัวสะกดสามารถกำหนดรหัสซาวน์เด็กซ์ได้ดังตารางที่ 4.3

ตารางที่ 4.3 รหัสซาวน์เด็กซ์ของพยัญชนะที่เป็นตัวสะกด

พยัญชนะที่ใช้ในตัวสะกด	รหัสซาวน์เด็กซ์
ก ข ค กร คร ฆ ต ฆ	ก
ง	ง
ม	ม
น ณ ร ญ ล ฬ	น
ว	ว
ย	ย
ฮ	ฮ
จ ช ซ ฎ ฏ ฐ ท ฒ ด ต ถ ท ธ ส ศ ษ ษฐ ตร ชร ทร ติ ตุ ทธิ ฒิ ทธ รถ	ต

พยัญชนะต้นสระ	รหัสซาวน์เด็กซ์
บ ป พ ฟ ภ	บ

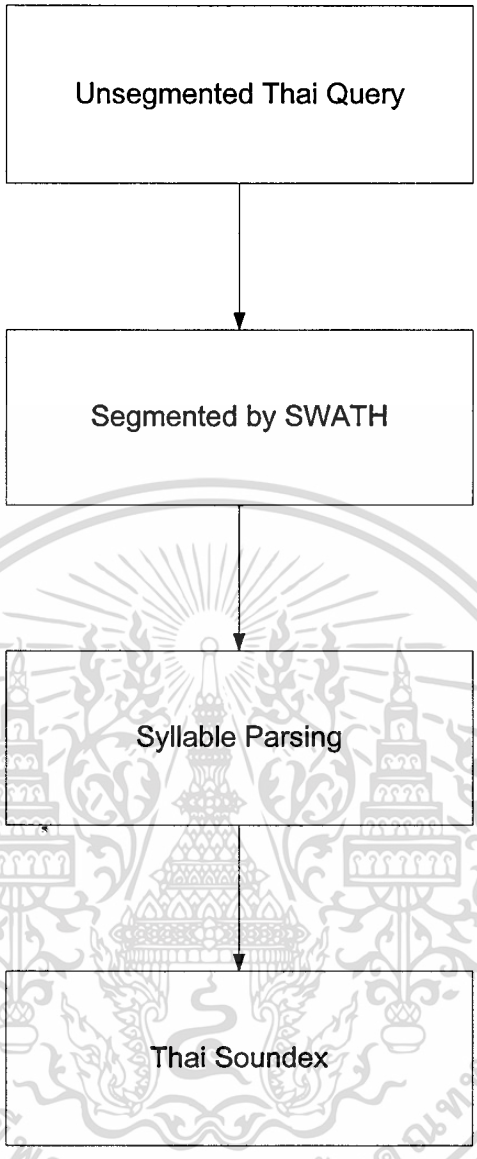
โดยมีตัวอย่างของคำที่ผ่านการสร้างรหัสซาวน์เด็กซ์เพื่อหาคำพ้องเสียงดังตารางที่ 4.4 คือ

ตารางที่ 4.4 ตัวอย่างคำที่เข้ารหัสซาวน์เด็กซ์

คำ	รหัสซาวน์เด็กซ์
บันได	บBนดH
บรรได	บBนดH
บันดัย	บBนดH
สัก	สBก
ศักดิ์	สBก
กิจ	กEต
กิตต์	กEต

การที่จะค้นหาคำพ้องเสียงให้ได้ผลลัพธ์ดังตารางซาวน์เด็กซ์ที่กล่าวมาข้างต้นนั้น ต้องขึ้นอยู่กับการตัดคำภาษาไทยและการแบ่งแยกพยางค์ของคำนั้นๆให้ถูกต้องด้วย จึงจะทำให้ผลลัพธ์ออกมาได้ผลลัพธ์ถูกต้องอย่างที่ต้องการ โดยในส่วนของการตัดคำนั้นได้ใช้โปรแกรม SWATH (Smart Word Analysis for THai) ซึ่งโปรแกรมนี้เป็นโปรแกรมสำหรับตัดคำภาษาไทย ที่พัฒนาโดยศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ และในส่วนของการแบ่งแยกพยางค์ของคำและส่วนของโปรแกรมแทนที่คำด้วยรหัสซาวน์เด็กซ์นั้นได้เขียนขึ้นโดยใช้ภาษา PERL

ขั้นตอนการทำงานของโปรแกรมค้นหาคำพ้องเสียงภาษาไทยโดยใช้วิธีการกำหนดรหัสซาวน์เด็กซ์ จากการทดลองมีขั้นตอนการทำงานดังแสดงในรูปที่ 4.2



รูปที่ 4.2 ขั้นตอนการทำงานของการสร้างรหัสซาวน์เด็กซ์

ภายหลังจากที่เราได้รหัสซาวน์เด็กซ์มาแล้วเราก็จะสามารถใช้รหัสซาวน์เด็กซ์ที่ได้ไปทำการเปรียบเทียบคำว่าเป็นคำที่พ้องเสียงกับคำที่ผู้ใช้งานต้องการค้นหาหรือไม่ ซึ่งในระบบสืบค้นข้อมูลภาษาไทยจะมี INDEX FILE ของคำไทยที่แปลงเป็นรหัสซาวน์เด็กซ์แล้วแยกอยู่ต่างหากจาก INDEX FILE ของคำที่ปรากฏอยู่ในเอกสารต้นฉบับที่ใช้ในการค้นหาข้อมูล

4.3 เครื่องมือและภาษาที่ใช้ในการพัฒนาระบบ

ในการพัฒนาระบบงานนี้ ได้ใช้เครื่องมือในการพัฒนาระบบงาน ดังนี้

4.3.1 ฮาร์ดแวร์ (Hardware)

ฮาร์ดแวร์ที่ใช้ในการพัฒนาระบบประกอบไปด้วยเครื่องคอมพิวเตอร์ต่อไปนี้

4.3.1.1 เครื่องเซิร์ฟเวอร์ สำหรับให้บริการสืบค้นข้อมูลภาษาไทย ผ่านเว็บเพจ โดยมีคุณสมบัติ ดังนี้

- CPU : Pentium III 500 MHz
- RAM : 128 MB
- Hard Disk : 6 GB
- NIC : 10/100 PCI

4.3.1.2 เครื่องไคลเอนต์ สำหรับการพัฒนาระบบ และทดสอบใช้งานระบบ โดยมีคุณสมบัติ ดังนี้

- CPU : Pentium III 1 GHz
- RAM : 384 MB
- Hard Disk : 20 GB
- NIC : 10/100

4.3.2 ซอฟท์แวร์ (Software)

ซอฟท์แวร์ที่ใช้ในระบบ มีดังต่อไปนี้

4.3.2.1 Apache Web Server บน Linux สำหรับให้บริการสืบค้นข้อมูลภาษาไทยผ่านเว็บเพจ

4.3.2.2 PostgreSQL สำหรับให้บริการเป็น Database Server

4.3.2.3 Linux สำหรับใช้เป็นระบบปฏิบัติการของเครื่องเซิร์ฟเวอร์

4.3.2.4 Windows XP สำหรับใช้เป็นระบบปฏิบัติการของเครื่องไคลเอนต์

4.3.2.5 SWATH สำหรับใช้เป็นโปรแกรมตัดคำภาษาไทย

4.3.2.6 BZCAT สำหรับใช้ดูข้อมูลที่โดนบีบอัดโดย BZIP2

4.3.2.7 PERL สำหรับใช้เป็นภาษาคอมพิวเตอร์ในการพัฒนาระบบ

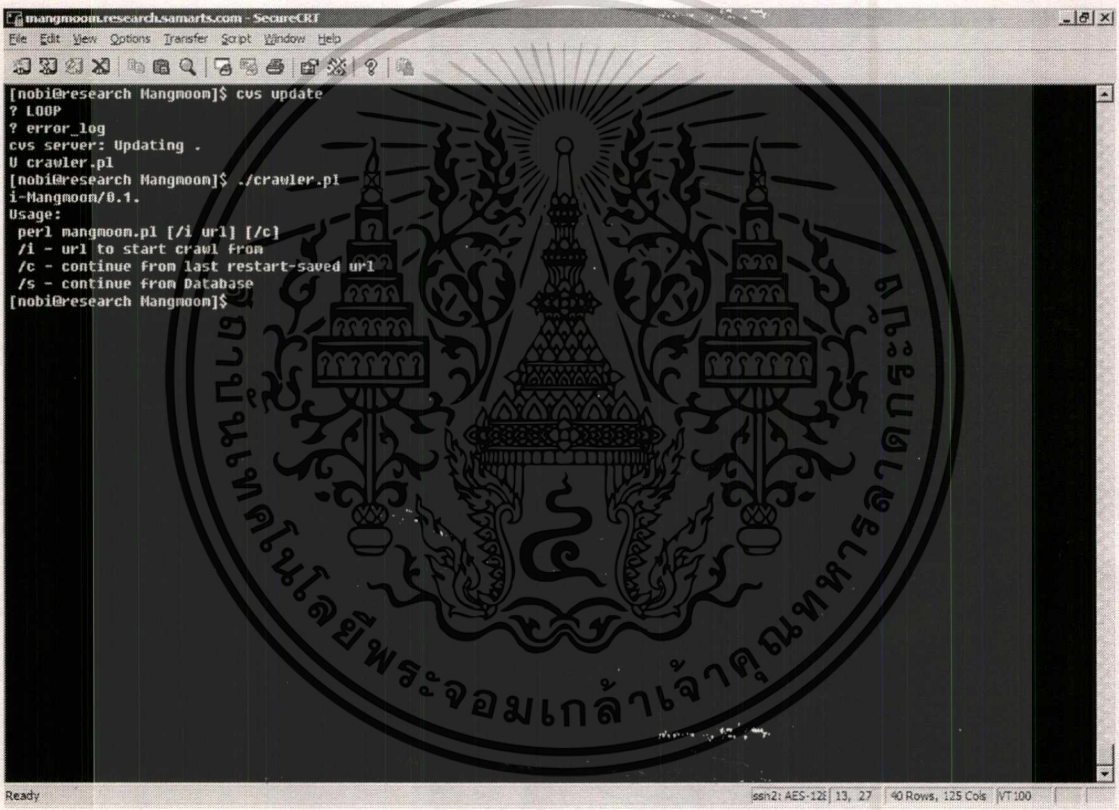
4.3.2.8 CVS สำหรับใช้งาน version control ในการพัฒนาโปรแกรม

4.4 รายละเอียดการพัฒนาระบบสืบค้นข้อมูลภาษาไทย

ระบบสืบค้นข้อมูลภาษาไทยที่ได้พัฒนานี้ ได้ออกแบบให้มีการทำงานแบ่งออกเป็น 2 ส่วน คือ ส่วนจัดเตรียมเอกสารสำหรับการค้นหา และ ส่วนติดต่อกับผู้ใช้โปรแกรมสืบค้นข้อมูล

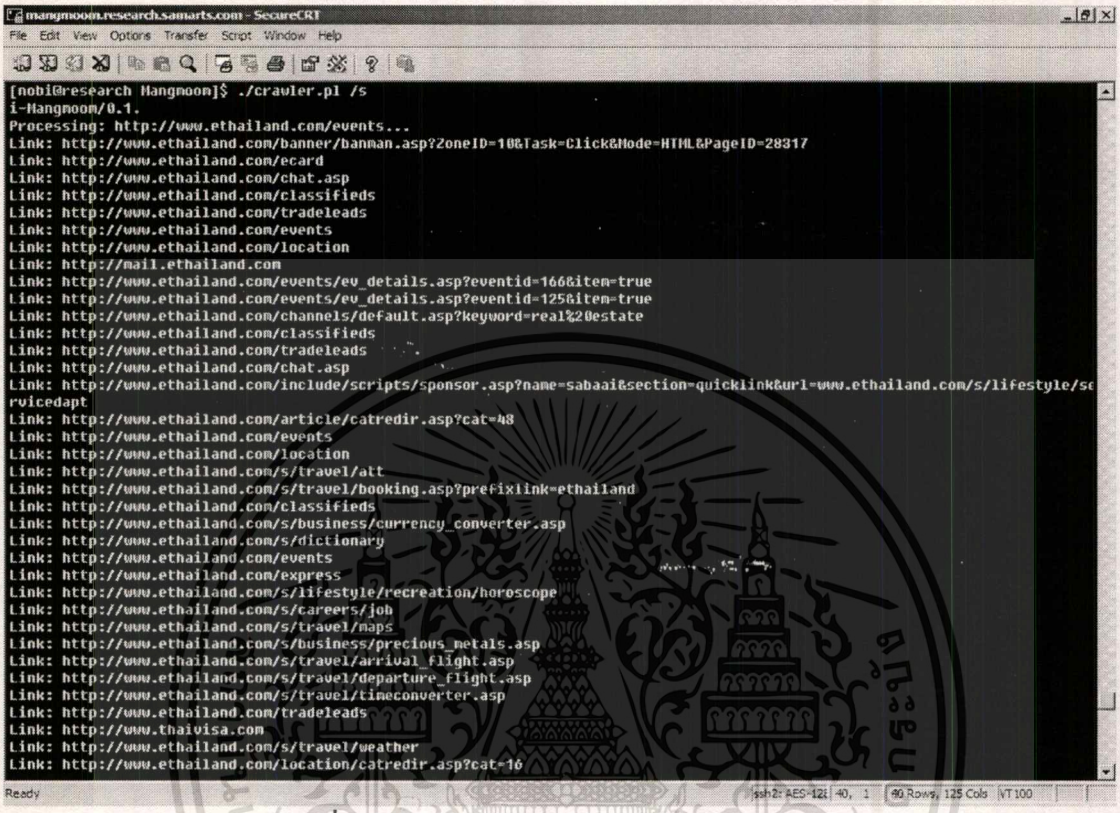
4.4.1 ส่วนจัดเตรียมเอกสารสำหรับการค้นหา

หน้าจอของโปรแกรมในส่วนจัดเตรียมเอกสารสำหรับการค้นหา จะเป็นหน้าจอที่มีการทำงานอยู่ใน text mode มีหน้าจอของตัว web crawler ดังรูปที่ 4.3

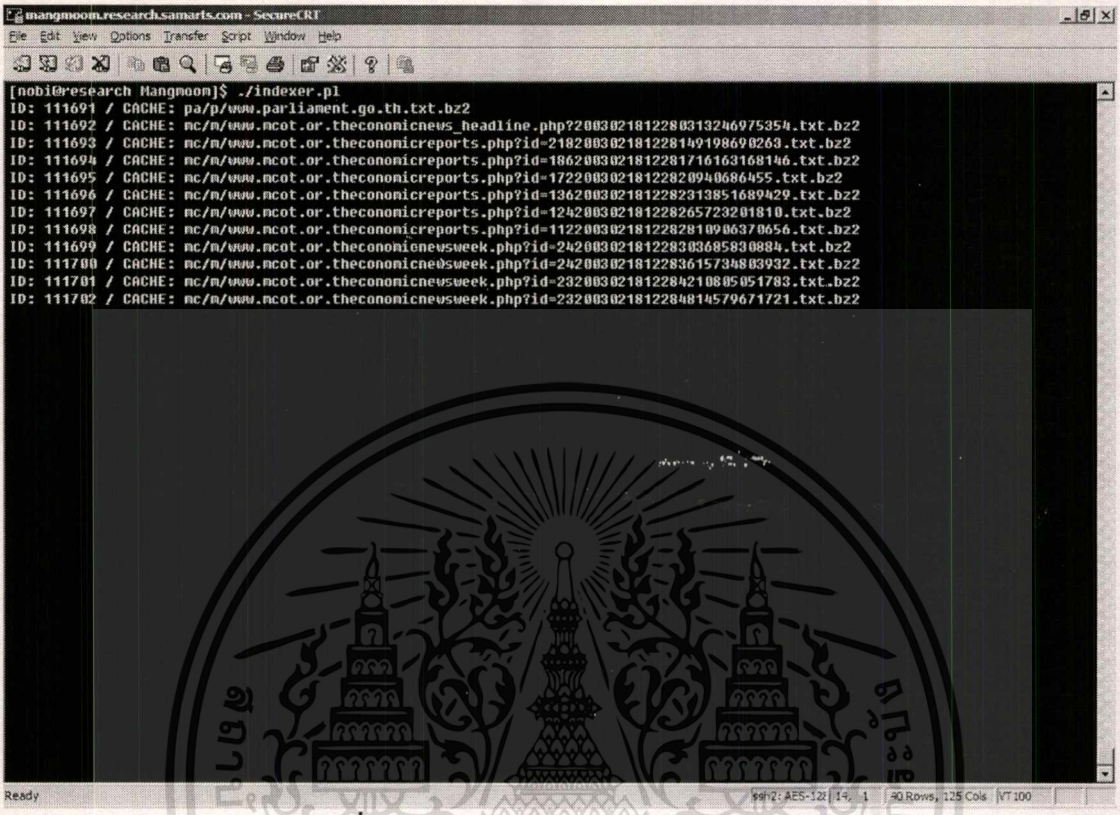


รูปที่ 4.3 หน้าจอของ web crawler

และมีหน้าจอเป็นดังรูปที่ 4.4 เมื่อสั่งให้โปรแกรม web crawler ทำงาน



รูปที่ 4.4 หน้าจอของ web crawler ในขณะทำงาน



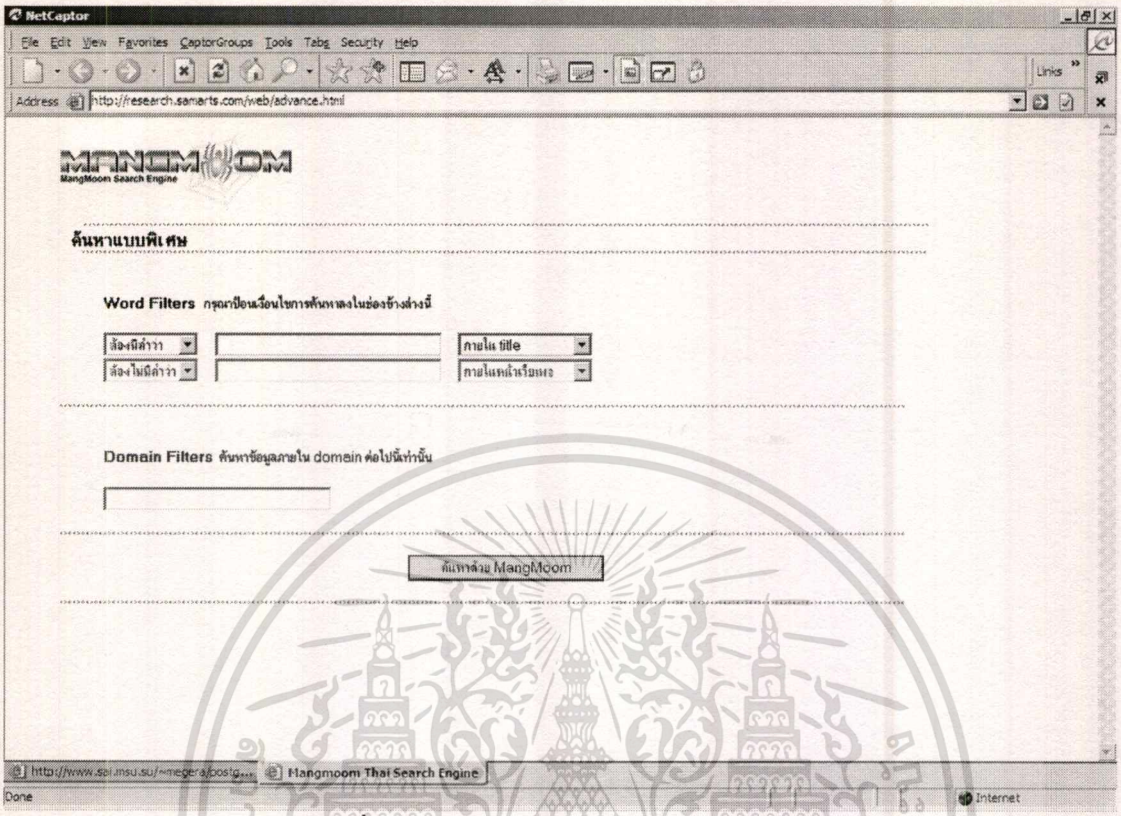
รูปที่ 4.5 หน้าจอของ indexer ในขณะทำงาน

4.4.2 ส่วนติดต่อกับผู้ใช้โปรแกรมสืบค้นข้อมูล

หน้าจอของส่วนติดต่อกับผู้ใช้โปรแกรมสืบค้นข้อมูลจะเป็นหน้าจอที่มีการติดต่อกับผู้ใช้งานผ่านทางเว็บเบราเซอร์ ประกอบไปด้วยหน้าจอดังต่อไปนี้

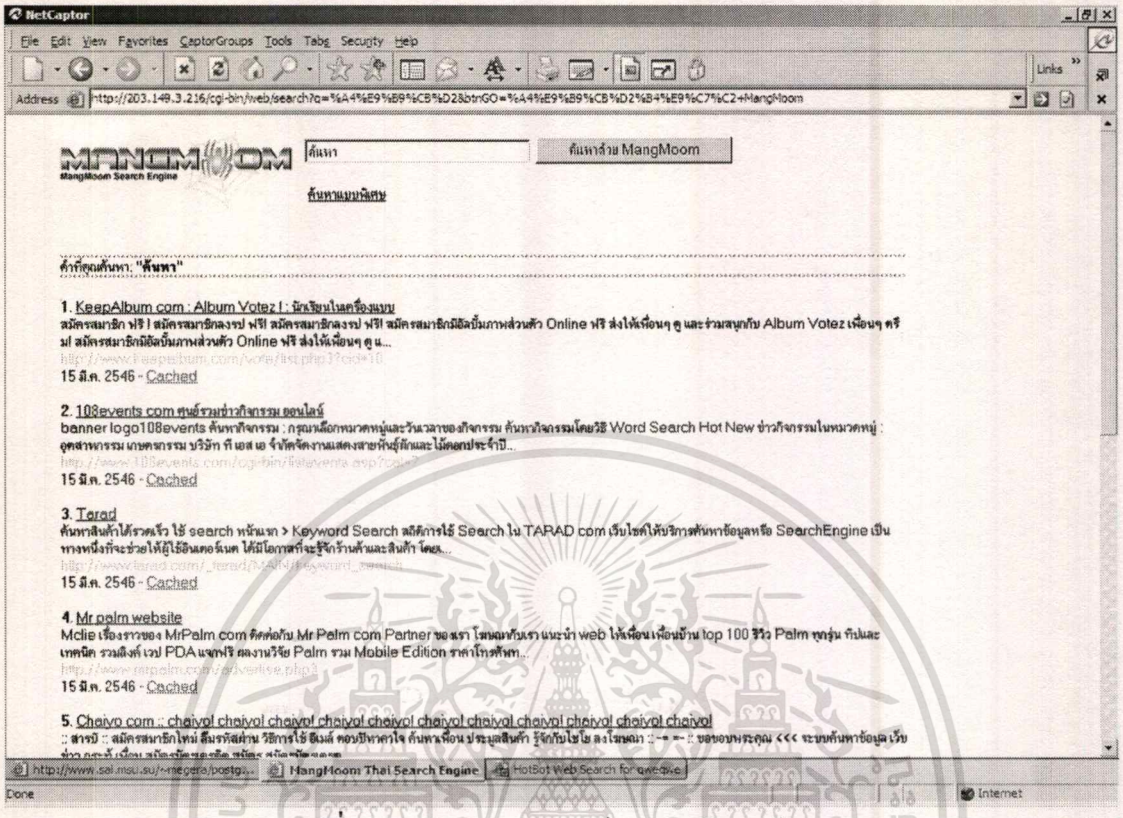
รูปที่ 4.6 หน้าจอโปรแกรมรอรับข้อมูลที่จะสืบค้น

จากรูปที่ 4.6 จะเป็นการค้นหาข้อมูลในแบบธรรมดา ไม่เจาะจงเฉพาะว่าให้หาข้อมูลเฉพาะตรงส่วน title ของเว็บเพจ หรือเจาะจงว่าให้หาข้อมูลเฉพาะในเว็บไซท์ที่ระบุ



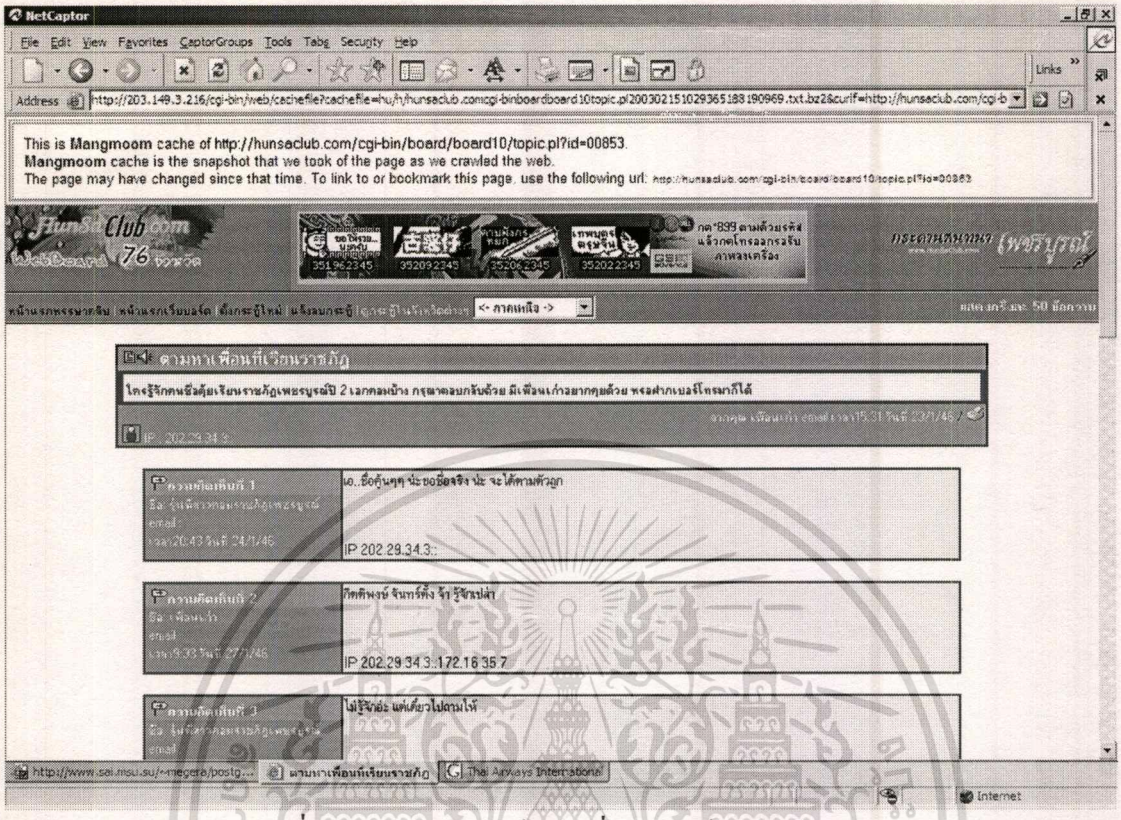
รูปที่ 4.7 หน้าจอการค้นหาข้อมูลในแบบพิเศษ

จากรูปที่ 4.7 ในการค้นหาแบบพิเศษจะสามารถกำหนดได้ว่าจะค้นหาข้อมูลแบบเจาะจงเฉพาะว่าให้หาข้อมูลเฉพาะตรงส่วน title ของเว็บเพจ หรือเจาะจงว่าให้หาข้อมูลเฉพาะในเว็บไซท์ที่ระบุ เป็นต้น



รูปที่ 4.8 หน้าจอการผลลัพธ์จากการค้นหาข้อมูล

จากรูปที่ 4.8 จะเป็นหน้าจอผลลัพธ์จากการค้นหาข้อมูลคำว่า “ค้นหา” ซึ่งหน้าจอผลลัพธ์จะเป็นหน้าจอแบบเดียวกันไม่ว่าจะค้นหาแบบพิเศษหรือค้นหาแบบธรรมดา



รูปที่ 4.9 หน้าจอแสดงเว็บเพจที่โดนจัดเก็บอยู่ใน cache

จากหน้าจอผลลัพธ์ของการค้นหาข้อมูล ผู้ใช้งานจะสามารถกดดู cache ของเว็บเพจที่ได้โดยหลังจากกดที่คำว่า cache โปรแกรมก็จะแสดงหน้าเว็บเพจที่จัดเก็บไว้ใน cache ดังรูปที่ 4.9

บทที่5
บทสรุปและวิจารณ์

5.1 บทสรุปและวิจารณ์

ในโครงการนี้ได้ดำเนินการ ศึกษาข้อมูลทฤษฎี และเทคโนโลยีด้านการสืบค้นข้อมูลในเครือข่ายอินเทอร์เน็ต และศึกษาคิดค้นวิธีสืบค้นข้อมูลภาษาไทยที่เป็นคำพ้องเสียง เพื่อช่วยให้การค้นหาข้อมูลภาษาไทยให้มีประสิทธิภาพมากยิ่งขึ้น เพื่อให้ผู้ค้นหาข้อมูลได้รับได้รับผลลัพธ์ที่ดีมากยิ่งขึ้น

5.2 ประโยชน์ที่ได้รับจากการพัฒนาระบบ

- ได้รับความรู้ในเทคโนโลยีใหม่ และเกิดทักษะ และความเข้าใจในการนำเทคโนโลยีใหม่มาประยุต์ใช้ในการพัฒนาระบบ
- ได้รับความรู้จากการค้นหาข้อมูลภาษาไทย เพื่อนำไปประยุกต์ใช้ในระบบบริการรูปแบบอื่นๆต่อไปในอนาคต
- ได้ความรู้ และทักษะในการพัฒนาระบบ และได้ระบบที่สามารถค้นหาข้อมูลที่เป็นคำพ้องเสียงภาษาไทยได้

5.3 ข้อจำกัดของระบบ

ระบบสืบค้นข้อมูลภาษาไทยที่พัฒนาขึ้นใหม่มีข้อจำกัดในบางประการดังนี้

- เนื่องจากการสร้างรหัสซาวน์เด็กซ์ที่นำเสนอในโครงการนี้จะได้ผลที่ถูกต้องได้จะต้องผ่านการตัดแบ่งพยางค์ภาษาไทยที่ถูกต้องเสียก่อน ดังนั้นหากการตัดแบ่งพยางค์ทำมาไม่ถูกต้องก็จะทำให้การค้นหาคำพ้องเสียงเกิดความผิดพลาดเนื่องจากได้รหัสซาวน์เด็กซ์ที่ไม่ถูกต้อง
- เนื่องจากการตัดแบ่งพยางค์ภาษาไทย และการสร้างรหัสซาวน์เด็กซ์ในโครงการนี้นั้นเกิดจากการคิดค้นขึ้นเองดังนั้นจึงไม่สามารถรับประกันได้ว่าจะถูกต้องแม่นยำ 100%

5.4 ข้อเสนอแนะ

ข้อเสนอแนะสำหรับผู้ที่ต้องการจะนำระบบไปศึกษาหรือนำไปพัฒนาต่อไปในอนาคต

- ควรขยายความสามารถของระบบให้สามารถค้นหาข้อมูลได้มากประเภทยิ่งขึ้น เช่น ค้นหาข้อมูลประเภทรูปภาพ
- ควรมีการออกแบบให้ระบบสามารถรองรับการใช้ในสภาวะที่มีผู้ใช้เข้ามาค้นหาข้อมูลพร้อมๆกันเป็นจำนวนมากได้
- ควรพัฒนาเทคนิคการตัดแบ่งพยางค์ภาษาไทยและการสร้างรหัสซาวน์เด็กซ์ต่อไปให้ดียิ่งขึ้น

บรรณานุกรม

Avi Rappaport. **Robots & Spiders & Crawlers: How Web and intranet search engines follow links to build indexes.** [Online]. Available: http://www.cs.uiowa.edu/~hshen/Robots.pdf

Sunny Lam. 2001. **The Overview of Web Search Engines.** Department of Computer Science University of Waterloo.

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายอรรถพล อาจไชยธร
วัน-เดือน-ปี เกิด	4 มีนาคม 2516
สถานที่เกิด	กรุงเทพฯ
วุฒิการศึกษาระดับปริญญาตรี	วิทยาศาสตรบัณฑิต(วิทยาการคอมพิวเตอร์)
สถานที่สำเร็จการศึกษา	มหาวิทยาลัยมหิดล
ปีการศึกษาที่สำเร็จการศึกษา	2538
ตำแหน่งหน้าที่ปัจจุบัน	System Analyst
สถานที่ทำงาน	บริษัท สามารถอินโฟมีเดีย จำกัด